Geoinformatica manuscript No.
(will be inserted by the editor)

Decentralized querying of topological relations between regions monitored by a coordinate-free geosensor network

Myeong-Hun Jeong · Matt Duckham

Received: date / Accepted: date

Abstract Geosensor networks present unique resource constraints to spatial computation, including limited battery power, communication constraints, and frequently a lack of coordinate positioning systems. As a result, there is a need for new algorithms that can efficiently satisfy basic spatial queries within those resource constraints. This paper explores the design and evaluation of a family of new algorithms for determining the topological relations between regions monitored by such a resource-constrained geosensor network. The algorithms are based on efficient, decentralized (in-network) variants of conventional 4-intersection and intersection and difference models, with in-network data aggregation. Further, our algorithms operate without any coordinate information, making them suitable applications where a positioning system is unavailable or unreliable. While all four algorithms are shown to have overall communication complexity $O(n)$ and optimal load balance $O(1)$, the algorithms differ in the level of topological detail they can detect; the types of regions they can monitor; and in the constant factors for communication complexity. The paper also demonstrates the impact of finite granularity observations on the correctness of the query results. In the conclusions, we identify the need to conduct further fundamental research on the relationship between topological relations between regions and limited granularity sensor observations of those regions.

Keywords geosensor network · decentralized spatial computing · 4-intersection model · intersection and difference model · granularity

Myeong-Hun Jeong
Department of Infrastructure Engineering, University of Melbourne, Victoria 3010, Australia
E-mail: mh.jeong@student.unimelb.edu.au

Matt Duckham
Department of Infrastructure Engineering, University of Melbourne, Victoria 3010, Australia
E-mail: mduckham@unimelb.edu.au

1 Introduction

Geosensor networks (wireless network of tiny, untethered computing devices monitoring phenomena in geographic space [22]) present exciting new possibilities to a broad range of environmental applications, like habitat monitoring, carbon accounting, and precision agriculture. However, the inherent resource constraints in geosensor networks, in particular limited energy resources, demand new approaches to spatial computing that can minimize communication and even operate without knowledge of nodes' coordinate locations.

The single most important technique for minimizing communication is the design of *decentralized* algorithms. In a decentralized algorithm, no node has global knowledge of the state of the entire network; instead each node has local knowledge about its state and that of its immediate neighbors. Decentralized algorithms help to minimize communication by ensuring information is communicated to neighbors only as required, and aggregated and processed within the network.

In addition to communication constraints, nodes frequently lack access to information about their coordinate position. Nodes may be located in environments that are unfavorable for positioning systems (e.g., using GPS in dense vegetation or underwater environments). Energy constraints may hamper positioning (e.g., where positioning systems have high energy budgets or long time-to-first-fix). Or positioning systems may simply be unavailable (e.g., low-cost, disposable nodes that simply lack sophisticated localization sensors). However, in such cases nodes are still expected to have access to one important piece of spatial information: neighborhood. Thus, although there is no information about the location of the sensors, it is possible to deduce topological relations between regions based on basic point-set topology concepts [4, 8, 9]. The topological aspects have been distinct from the geometric aspects [1]. The topology (i.e., qualitative geometry [14]) therefore is another way of determining topological relations not based on computational geometry.

Accordingly, this research focuses on efficient, decentralized, and coordinate-free algorithms for determining the topological relations between multiple regions monitored by such a resource-constrained geosensor network. The algorithms adapt two conventional models of the topological relations between regions [4, 8] to the unique constraints of decentralized spatial computing environments. The algorithms are kept efficient through the combination of two complementary decentralized computing strategies: restricting computation to spatial structures (boundaries) to reduce the number of nodes that have to take part in communication; and data aggregation to eliminate the communication of redundant information. This paper does, however, substantially revise and extend our previous work presented in a recent short conference article [6].

Following a review of related work (section 2), section 3 precisely specifies the construction of four related algorithms in order of increasing sophistication. Section 4 then presents an experimental evaluation of the overall efficiency, load balance, and accuracy of the algorithms, using simulation. Finally, the paper concludes with a discussion of the limitations of the approach and future

work, in particular the importance of granularity effects in accurate topological queries (section 5).

2 Background

2.1 Decentralized spatial computing

The design of decentralized algorithms in wireless sensor networks is a significant and active research problem. In-network data aggregation, for example, is one fundamental technique for reducing the number of data transmissions in decentralized algorithms (e.g., [12, 17, 19, 34, 40]). The algorithms in this research also depend partly on data aggregation.

However, in designing decentralized *spatial* algorithms, the primary focus has been on using spatial structures, like boundaries, regions, Voronoi cells, or planar communication graphs, for structuring communication and computation. For example, [29, 35] use decentralized plane sweeps to efficiently coordinate spatial queries, including identifying peaks, pits, and saddle points in monitored fields. [31] use Voronoi diagrams to extend existing in-network aggregation techniques in order to perform spatial averages, weighted by the size of Voronoi cells for each node. [18] show how energy requirements for spatiotemporal queries can be substantially reduced by using the boundaries of regions to report for the entire region.

This paper continues the approach of using spatial structures (in our case region boundaries) to efficiently compute topological relations between regions. Specifically, this paper is concerned with queries about the topological relations between two spatial regions, such as "Does region A *cover* or *contain* region B?" The regions in question may be bona fide (such as the presence or absence of monitored pollutant) or may be fiat regions derived from thresholding continuous fields (e.g., temperatures above 30°C) [39]. For example, specific applications of these algorithms might include whether regions of high fuel load and temperature hot-spots overlap in bushfire monitoring; or whether regions of high nitrogen update are contained within regions of high soil moisture in conservation wetlands creation.

2.2 Topological relations between regions

There are several related models of the topological relations between spatial regions in artificial intelligence and spatial information science. The most well-known is the 4-intersection model [8], which considers the intersections between the two regions' (more specifically, point sets') topological interiors($^\circ$) and boundaries(∂). For example, topological relations between region A and region B can be denoted by a four-tuple ($\partial A \cap \partial B$, $\partial A \cap B^\circ$, $A^\circ \cap \partial B$, $A^\circ \cap B^\circ$). By assigning the values empty ($\emptyset$) and non-empty ($\neg\emptyset$) for the four intersections, this model can differentiate $2^4 = 16$ binary topological relations. But eight of

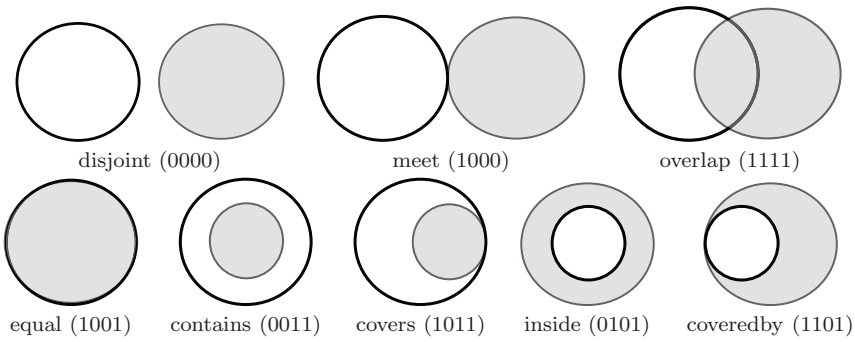


Fig. 1 The eight topological relations between two regions based on the 4-intersection model

these sixteen relations can be realized for two simple regions in the plane as shown in Fig.1.

Furthermore, [9] adds to the 4-intersection model the two objects' complements, yielding the 9-intersection model (although the resulting topological relations are identical to the case of connected regions with Jordan boundaries). An alternative formulation in [4] uses intersection and difference between regions to improve the efficiency of computing these relations.

These approaches assume regions are "simple" (homeomorphic to a disk). However, other extensions for describing the topological relations between complex areal objects, such as disconnected regions and regions with holes, have also been proposed (e.g., [10, 16]). The TRCR model (topological relations for composite regions) [2] provides a mechanism for describing topological relations between *composite* regions (i.e., composed from multiple disconnected simple regions) based on the 4-intersection model. The research by [21] extends the approach further for handling of the compositions of disconnected regions and holes. [30] has also introduced definitions of general spatial data types for complex regions based on the 9-intersection model by using a proof technique called proof-by-constraint-and-drawing. Based on this model, 33 different topological relations can be identified between two complex region objects and these topological relations are grouped by generic topological cluster predicates (i.e., the familiar set of eight topological relations).

However, these models are not appropriate to be adopted directly in geosensor networks [5]. Specifically, from a decentralized perspective, while an individual node may be able to determine if the intersection between two point sets *is not* empty (e.g., a single node that locally senses both region A and region B can locally infer that $A \cap B \neq \varnothing$), it is never possible for an individual node to determine if the intersection between two point sets *is* empty (e.g., a single node that does not locally sense both regions A and B can not infer that $A \cap B \neq \varnothing$, since there may be some other node in the network that can sense both region A and region B). This observation has direct implications for designing efficient decentralized spatial algorithms, when compared with conventional, centralized alternatives.

It is worth mentioning a completely different, axiomatic approach that can also be used to describe topological spatial region relations, including disconnected regions and regions with holes. Region connection calculus (RCC) [3, 23] relies on a binary *connectedness* relation, rather than boundaries, interiors, and complements of point sets. In our work, the point-set based approach is preferred to RCC, because boundaries, interiors, and complements can be efficiently and locally computed in a geosensor network (e.g., [6, 7, 11, 39]). By contrast, connectedness is a primitive in RCC, and practically computing whether two nodes are "connected" in a sense compatible with RCC (e.g., path connected) would be highly inefficient in a geosensor network.

While computational geometry can be used to determine topological relations, we regard topological relations as set-theoretic constructs. None of these approaches mentioned above [2, 8, 9, 23] use any geometry at all in defining topological relations, even though they all have well-defined boundary and interior concepts.

Finally, unlike most other recent work on decentralized spatial algorithms for topological queries (e.g., [7, 11, 15, 27, 39]), this paper does not assume any knowledge of coordinate positions for nodes. One of our previous works has begun to address the issue of detecting topological *changes* to region relations (e.g., transitioning from meet to overlap) without coordinate information [13]. By contrast, in this paper we are concerned with the complementary question of *static* topological relations between regions (i.e., "snapshot" queries). Similarly, this research extends the previous work [6] such that this paper can generate topological relations between simple and complex regions ranging from the coarse granularity topological relations to the fine granularity topological relations at a small computational cost.

3 Algorithms

In this section we present four algorithms, in order of increasing complexity. The first algorithm represents a naïve approach, where all nodes at the intersection of two regions A and B are actively involved in the computation. The second algorithm improves efficiency by restricting computation only to nodes at the boundary of the intersection of A and B, but at the cost of reduced topological relation granularity. The third algorithm goes further and involves the one-hop neighbors of nodes at the boundary of the intersection of A and B in order to regain the fine topological granularity at a small computational cost. Finally, the fourth algorithm demonstrates how the approach can be extended to determine the topological relations between complex areal objects composed of multiple disconnected parts.

It is worth noting that even though we can use computational geometry to compute topological relations, set-theoretic constructions are more fundamental. This is important in the context of geosensor networks, where coordinate location is either resource intensive to compute, or frequently unavailable.

Thus, algorithms presented in this paper propose qualitative approach based on relative neighborhood information to determine topological relationship.

3.1 Algorithm preliminaries

The design and specification of the algorithms presented in this paper follow the approach of Santoro [28]. Each algorithm begins with a list of restrictions, specifying the assumptions required for the algorithm to operate. Following previous work, in this paper we assume as restrictions:

- a geosensor network, modeled as a connected, undirected graph $G = (V, E)$. Our algorithm places no restrictions on whether the graph is planar, although the results of the algorithm may vary depending on the network structure (cf. [26]). The neighbors of a node $v \in V$ are denoted $nbr(v)$, where $nbr(v) = \{v' | v, v' \in E\}$.
- each node has sensors capable of determining whether a node detects a region A and/or B, modeled as a function $sense : V \rightarrow \mathscr{P}(\{A, B\})$ ($\mathscr{P}(\{A, B\}$ is the power set of $\{A, B\}$).
- communication is *reliable*, in the sense that all messages sent will be delivered without corruption in a finite amount of time. We make no assumptions about the communication latency or the order in which messages are received (e.g., individual messages may be delayed or even "overtake" other messages).

The neighborhood of each node (i.e., $nbr(v)$) is simply those nodes with which it can engage in direct (one-hop) communication. Thus, it is guaranteed that each node knows its neighbors because they must be within communication range, in which case the reliable communication assumption ensures that node will hear messages from them. Accordingly, nodes must already know which neighbors are in their immediate (one-hop communication) vicinity. This is a standard assumption in distributed systems [28].

After the restrictions, each algorithm specifies a set of states for nodes (written in small capital letters); the allowable transitions between states (together forming a state transition system); and the initial states for nodes. In this paper, all the algorithms are initialized with all nodes in state IDLE, except one designated sink node initialized in the SINK state. The sink node is responsible for initiating the query and collating the partially processed responses from targeted nodes in the network. Although we use a single sink node in our algorithm, the algorithm might easily be extended to operate with multiple or zero sink nodes (in the latter case retaining information in the network for use in subsequent, more sohisticated algorithms).

Following [28], each node has capabilities including access to local memory, local processing, and communication. Local memory includes the initially defined states (see above). It may also contain *local variables*: data structures that each node may use to store information created during algorithm execution (listed as the last item in the algorithm header).

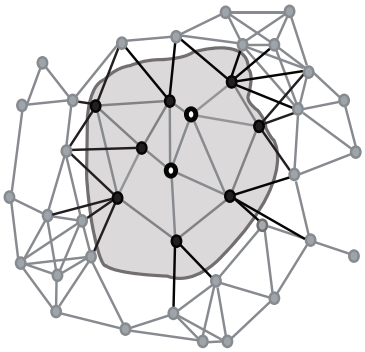


Fig. 2 Boundary and interior nodes in a sensed region (Black nodes are boundary nodes and white nodes with black thick stroke are interior nodes)

In each state, a node can only respond to system events (written in italics). Just two types of events are required in this paper[1]: receiving a message (*Receiving* keyword) or spontaneous events (*Spontaneously* keyword, typically used to start an algorithm). When an event occurs, a node will react by executing an indivisible and terminating sequence of operations called an *action*. Actions are executed without interruption (i.e., no other events can affect an action) and must end within a finite amount of time. For each state and event there must be exactly one action, State $\times$ Event $\rightarrow$ Action; state, event pairs with no specified actions are defined to be associated with the empty action.

Inside each action, it is important to be able to distinguish each individual node's *local* knowledge from the state of the network (i.e., a given node v will have access to its own sensed data $sense(v)$, but not to that of any other node $sense(v')$ unless it has previously been explicitly communicated to and locally stored at v). To enforce and highlight the local knowledge of a node, the over-dot notation $\mathring{sense}$ (termed "local" or "my" *sense*) is used to refer to the current node's knowledge of that function (i.e., for an arbitrary node $\circ \in V$ clear from the context, $\mathring{sense}$ is equivalent to $sense(\circ)$).

Given this information, each node can determine whether they are boundary nodes or interior nodes in a sensed region based on short-range, peer to peer communication. Fig.2 illustrates how to determine the boundary of the region in a geosensor network. Boundary nodes that can sense they are inside a sensed region and have a one-hop neighbor outside the region. Similarly, interior nodes are inside a sensed region, but only can communicate with nodes in a sensed region. Apparently, the boundary defined by geosensor networks will be necessarily at a coarser granularity than the phenomenon itself. However, this will be the case for any geographic information, which must also necessarily be at a limited level of granularity (in indeed the granularity limitations may actually be lessened by geosensor networks, where typically nodes are spatially densely distributed).

[1] [28] define a third system event type: an alarm/trigger event.

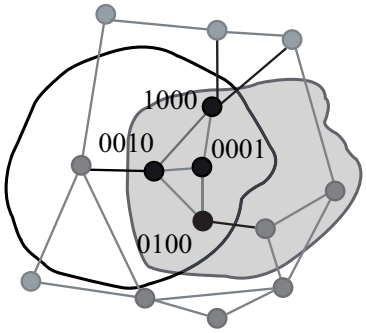


Fig. 3 Four inferred bit combinations at the intersection of two regions, A and B. 1000 is in $\partial A \cap \partial B$; 0100 is in $\partial A \cap B^\circ$; 0010 is in $A^\circ \cap \partial B$; 0001 is in $A^\circ \cap B^\circ$

Using this idea, it is possible to adopt the 4-intersection model in order to systematically determine topological relations between regions monitored by a geosensor network. Later sections will introduce some more sophisticated algorithms with simplified figures.

3.2 Basic algorithm

Algorithm 1 (basic) is a direct, decentralized analog of the well-known 4-intersection model [8]. Starting with the sink node, the network is flooded with a `ping` message, as the basis for constructing a routing tree (with each node storing its immediate parent in the tree, $\mathring{parent}$). This step requires in total $|V|$ messages, leading to $\Theta(n)$ overall communication complexity (`ping` messages sent), with optimal $\Theta(1)$ load balance (number of messages per node). The payload of each `ping` message includes a node's sensed value (in or out of region A and/or B). Using this information about its neighbors, a node can deduce the following (see Fig. 3):

- A node that senses A and B ($\mathring{sense} = \{A, B\}$) and has at least one one-hop neighbor that senses both A and B, is at the boundary of A and B, i.e., $\partial A \cap \partial B \neq \varnothing$ (see Algorithm 1, line 17);
- A node that senses A and B and has at least one one-hop neighbor that senses B only, is also at the boundary of A, i.e., $\partial A \cap B^\circ \neq \varnothing$ (see Algorithm 1, line 18);
- A node that senses A and B and has at least one one-hop neighbor that senses A only, is also at the boundary of B, i.e., $A^\circ \cap \partial B \neq \varnothing$ (see Algorithm 1, line 19); and
- A node that senses A and B, and has only neighbors that also sense A and B is in the interior of A and B, i.e., $A^\circ \cap B^\circ \neq \varnothing$ (see Algorithm 1, line 21).

Each node stores the non-empty intersections it can deduce (Fig. 3) as a four-bit number, using the function $bnum : V \to \mathbb{B}^4$ (see Algorithm 1, line 4).

In turn, each non-zero bit signifies a node has deduced: $\partial A \cap \partial B$, $\partial A \cap B^\circ$, $A^\circ \cap \partial B$, $A^\circ \cap B^\circ$, respectively (e.g., see Fig. 3).

Algorithm 1 Decentralized 4-intersection algorithm for querying topological relations between regions A and B (basic)

1: Restrictions: reliable communication; connected, undirected graph $G = (V, E)$; $nbr : V \to \mathscr{P}(V)$ where $nbr(v) \mapsto \{v' | \{v, v'\} \in E\}$; $sense : V \to \mathscr{P}(\{A, B\})$
2: State Trans. Sys.: $\langle \{\text{SINK}, \text{INIT}, \text{IDLE}, \text{DONE}\}, \{(\text{INIT}, \text{SINK}), (\text{IDLE}, \text{DONE})\}\rangle$
3: Initialization: All nodes in state IDLE, except one node in INIT
4: Local variables: $bnum : V \to \mathbb{B}^4$, initialized $\mathring{bnum} := 0000$; $parent : V \to V \cup \{\varnothing\}$, initialized $\mathring{parent} := \varnothing$; visited neighbors N, initialized $N := \varnothing$

INIT
5: *Spontaneously*
6: **broadcast** (`ping`, $\mathring{sense}$) //*Sink initiates algorithm*
7: **become** SINK

SINK
8: *Receiving* (`rprt`, b)
9: **set** $\mathring{bnum} := b \vee \mathring{bnum}$
10: Deduce topological relations between A and B from $\mathring{bnum}$ according to Table 1

IDLE
11: *Receiving* (`ping`, x) from v
12: $N := N \cup \{v\}$ //*Update list of visited nodes*
13: **if** $\mathring{parent} = \varnothing$ **then** //*Check for first* `ping` *received*
14: **set** $\mathring{parent} := v$ //*Store tree parent*
15: **broadcast** (`ping`, $\mathring{sense}$) //*Continue building tree*
16: **if** $\mathring{sense} = \{A, B\}$ **then**
17: **if** $x = \varnothing$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 1000$ //$\partial A \cap \partial B \neq \varnothing$
18: **if** $x = \{B\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0100$ //$\partial A \cap B^\circ \neq \varnothing$
19: **if** $x = \{A\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0010$ //$A^\circ \cap \partial B \neq \varnothing$
20: **if** $N = \mathring{nbr}$ **then** //*Check if tree received from all neighbors*
21: **if** $\mathring{bnum} \neq 0000$ and $\mathring{sense} = \{A, B\}$ **then set** $\mathring{bnum} := 0001$//*Check for* $A^\circ \cap B^\circ$
22: **if** $\mathring{bnum} \neq 0000$ **then send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Initiate message to sink*
23: **become** DONE

DONE, IDLE
24: *Receiving* (`rprt`, b)
25: **if** $b \vee \mathring{bnum} \neq \mathring{bnum}$ **then** //*Check for new data*
26: **set** $\mathring{bnum} := b \vee \mathring{bnum}$ //*Data aggregation*
27: **send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Forward aggregate data*

When an IDLE node has received `ping` messages from all its neighbors, it transitions to a DONE state. If that node's inferred $\mathring{bnum} \neq 0000$ (i.e., if the node is somewhere at the intersection between A and B) it will forward a `rprt` message to the sink node (see Algorithm 1, lines 20–23). Nodes receiving a `rprt` message compare their existing knowledge with the received information using a logical, bitwise disjunction operator (see Algorithm 1, lines 25–27). Data aggregation is used to ensure only new information, not previously known to the node, is forwarded further towards the sink. This process of data aggregation is shown in Fig. 4. The bold dark edges are in the rooted tree, established using `ping` messages to return information to the sink node. In Fig. 4 step 1,

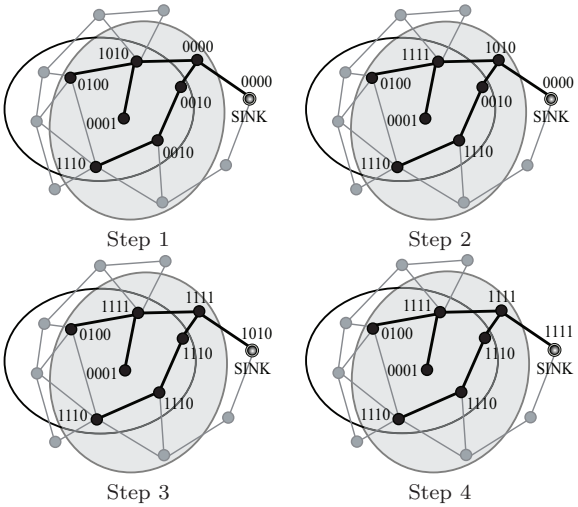


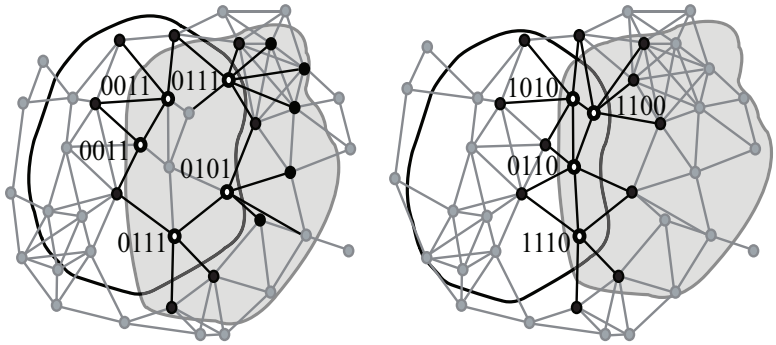
Fig. 4 Data aggregation process (steps 1–4)

Table 1 Determining the topological relation between regions (see basic algorithm 1, line 10 and Algorithm 3, line 10), for $bnum(v) \in \mathbb{B}^4$ of sink node v $\in V$. 1000 indicates $\partial A \cap \partial B \neq \varnothing$; 0100 indicates $\partial A \cap B^\circ \neq \varnothing$; 0010 indicates $A^\circ \cap \partial B \neq \varnothing$; and 0001 indicates $A^\circ \cap B^\circ \neq \varnothing$

$bnum(s)$ for sink node $s \in V$	Topological relation
0000	A, B disjoint
1000, 1100, 1110, 0110, 1010	A, B meet
1111, 0111	A, B overlap
1001, 1000	A, B equal
0011, 0010	A contains B
1011, 1010	A covers B
0101, 0100	A inside B
1101, 1100	A coveredby B

every node senses their own value and then updates their bit sequence. After receiving messages from all one-hop neighbors, every node with any non-zero bits forwards an `rprt` message to its parent node in the routing tree (Fig. 4 step 2). Before forwarding an `rprt` message, nodes compare their bit sequence with previously received information, and only forward new, previously unseen information to the sink node in Fig. 4, steps 3–4. Because of this data aggregation, each node may transmit at most four `rprt` messages, leading again to an overall worst case communication complexity of O(n) and load balance O(1). However, in practice, only a small proportion of nodes are expected to lie at the intersection of A and B (if any), meaning the average case communication complexity for `rprt` messages may be much better.

In the final step, the sink node can deduce the topological relation between the two regions using the bitwise disjunction of all received messages (Algorithm 1, line 10 and Table 1). Because of the limited spatial granularity of the sensor network, the table is *not* the same as the familiar 4-intersection model,



a. Overlap b. Meet

Fig. 5 Example bit combinations that do not directly correspond to the 4-intersection model: a. overlap relation and b. meet relation

but includes the full 16 possible bit combinations. For example, overlap is not simply 1111, but also 0111, because for two overlapping regions it is possible that *no* node in the neighborhood of a node that senses both A and B happens to sense *neither* A nor B (i.e., all neighbors of nodes that sense A and B, sense either A, B, or A and B). Fig. 5a illustrates an example of such an overlap configuration. Another example of granularity effects is the meet relation, where bit sequences includes 1100, 1110, 0110, and 1010 in addition to the expected 1000. Again, because of limited spatial granularity, a node that is at the boundary of A and B, may detect the boundary of A, the boundary of B, and the combined boundary of A and B through separate interactions with neighbors (see Fig. 5b).

Further, not all bit combinations specify a unique topological relation. Specifically, 1000 may indicate meet *or* equal; 1010 may indicate meet *or* cover; and 1100 may indicate meet *or* covered by. Fig. 6 illustrates these degenerate cases. Lastly, one of the sixteen topological spatial relations, 0001 (omitted from Table 1), can only occur where the boundaries of both regions are *beyond* the spatial extents of the network (as shown in Fig. 7).

3.3 Three-bit coarse resolution topological relation model between simple regions (3bit)

Algorithm 1 illustrates a number of issues that arise when decentralizing the 4-intersection model, and more importantly the confounding effects of the limited spatial granularity of a geosensor network. In particular, those relations that involve boundary conditions (meet, covers, and covered by) are in some senses not well-defined in that boundaries in a sensor network cannot be directly sensed; instead they must be inferred from pairs of neighbor nodes that straddle the boundary. Consequently, the granularity of the network places limitations on the level of detail that can be provided about the boundary location. However, limited granularity is a feature of any spatial data cap-

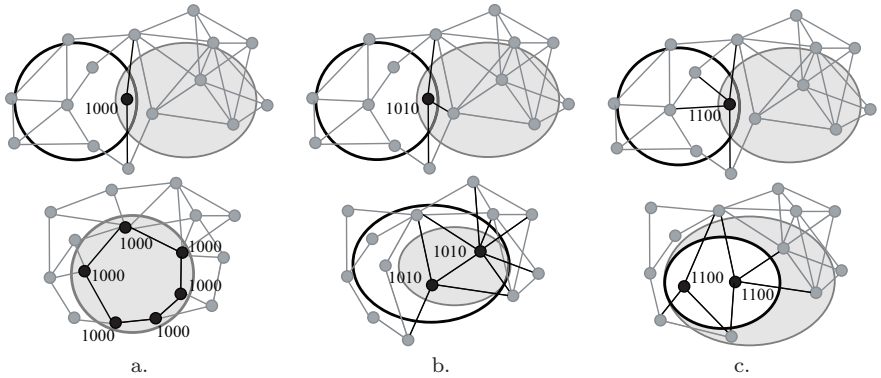


Fig. 6 Limited granularity and topological relation uniqueness: a. 1000 could be classified as meet or equal; b. 1010 could be classified as meet or cover; and c. 1100 could be classified as meet or covered by.

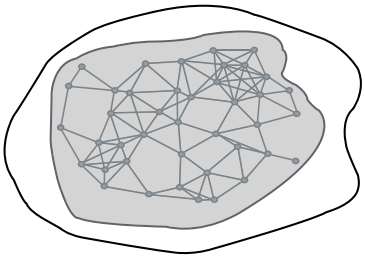
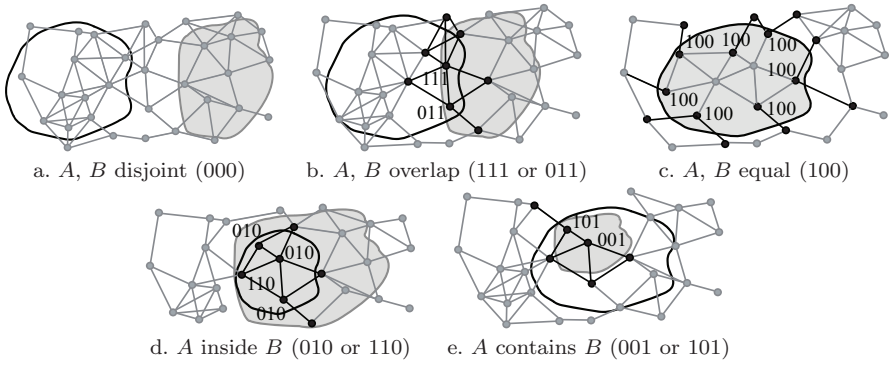


Fig. 7 The spatial extent of the network is smaller than the boundaries of both regions.

ture technology, and indeed a geosensor network might even expect to be at finer granularity than more traditional spatial data capture methods. Thus an alternative to address this issue is to move to a coarser level of topological granularity, instead distinguishing between just five (rather than the conventional 8) topological relations: disjoint, overlap, contains, inside, and equals.

Monitoring these coarser-grained topological relations can be achieved using just three bits (i.e., $\partial A \cap \partial B \neq \varnothing$, $\partial A \cap B^\circ \neq \varnothing$, $A^\circ \cap \partial B \neq \varnothing$) and involving only those nodes at the boundary of the intersection between A and B (see Fig. 8). Algorithm 2 (3bit) is a direct adaptation of Algorithm 1 using only these three-bit sequences. Crucially, the algorithm is expected to be computationally more efficient, requiring at most three `rprt` messages per node. Further, since `rprt` messages are only initiated at the *boundary* of the intersection between A and B (rather than over the *entirety* intersection of A and B), the overall number of `rprt` messages sent is expected to scale in proportion to $|V|^{\frac{D}{2}}$, where $D \in [1, 2)$ is the fractal dimension of the region [7]. Estimates of fractal dimension for geographic shapes vary widely. For example, fractal dimensions of regions representing urban growth are typically in the interval $D \in [1.2, 1.7]$ [32]; other features like river networks have higher estimated fractal dimensions, $D \in [1.4, 1.9]$ [25, 36]. However, irrespective of the precise



a. A, B disjoint (000) b. A, B overlap (111 or 011) c. A, B equal (100)

d. A inside B (010 or 110) e. A contains B (001 or 101)

Fig. 8 Three-bit coarse resolution topological relations between two spatial regions

value of D, involving only nodes at the boundary of regions in initiating `rprt` messages reduces the overall communication complexity for these messages to $O(n^k)$ where $0.5 \leq k < 1$. The reason behind this is that a fractal dimension is an index characterizing geometric forms of spatial objects (e.g., irregularity, scale-independence, and self-similarity) [32]. If spatial objects have Euclidean geometric regularity, the fractal dimension equals topological dimension. For example, if D equals 0, 1 or 2, it describes points, lines or surfaces respectively. Thus, a fractal dimension can be explained as a set which exceeds the topological dimension [20]. Accordingly, the fractal dimension of the boundary of the region component is $D \in [1, 2)$ [7].

3.4 Four-bit fine-grained topological relation between simple regions

Algorithm 3 (4bit) again returns to fine-granularity topological relations and a four-bit representation. However, building on the idea introduced in the previous section (that average-case efficiency can be improved by only involving nodes at the *boundary* of the intersection between A and B in the generation of `rprt` messages) Algorithm 3 uses a node's knowledge of its neighboring nodes to ensure only boundary nodes and any of their one-hop neighbors in the interior of the intersection of A and B send `rprt` messages to the sink. The intuition here is that nodes that are both in the interior of the intersection between A and B and have no neighbors at the boundary of the intersection of A and B hold only redundant information about the topological relation between the region (and so need never initiate an `rprt` message unlike Algorithm 1). Thus, Algorithm 3 is able to achieve the same fine-granularity topological detail as Algorithm 1, but with only nodes at most one-hop from the boundary of the intersection of A and B responding, again ensuring the number of `rprt` messages should be proportional to $|V|^{\frac{D}{2}}$, and so overall communication complexity of `rprt` messages $O(n^k)$, $0.5 \leq k < 1$.

Algorithm 3 can be summarized by highlighting the following features:

Algorithm 2 Querying coarse resolution topological relations between regions A and B (3bit)

Restrictions: reliable communication; connected, undirected graph $G = (V, E)$; $nbr : V \to \mathscr{P}(V)$ where $nbr(v) \mapsto \{v' | \{v, v'\} \in E\}$; $sense : V \to \mathscr{P}(\{A, B\})$
State Trans. Sys.: $\langle \{$SINK, INIT, IDLE, BNDY, DONE$\}, \{$(INIT, SINK), (IDLE, BNDY), (IDLE, DONE), (BNDY, DONE)$\}\rangle$
Initialization: All nodes in state IDLE, except one node in INIT
Local variables: $bnum : V \to \mathbb{B}^3$, initialized $\mathring{bnum} := 000$; $parent : V \to V \cup \{\varnothing\}$, initialized $\mathring{parent} := \varnothing$; visited neighbors N, initialized $N := \varnothing$

INIT
Spontaneously
  broadcast (`ping`, $\mathring{sense}$) //*Sink initiates algorithm*
  become SINK

SINK
Receiving (`rprt`, b)
  set $\mathring{bnum} := b \vee \mathring{bnum}$
  Deduce topological relation between A and B from $\mathring{bnum}$ according to Fig. 8

IDLE
Receiving (`ping`, x) from v
  $N := N \cup \{v\}$ //*Update list of visited nodes*
  if $\mathring{parent} = \varnothing$ **then** //*Check for first* `ping` *received*
    set $\mathring{parent} := v$ //*Store tree parent*
    broadcast (`ping`, $\mathring{sense}$) //*Continue building tree*
  if $x \neq \mathring{sense}$ and $\mathring{sense} = \{A, B\}$ **then**
    if $x = \varnothing$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 100$ //$\partial A \cap \partial B \neq \varnothing$
    if $x = \{B\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 010$ //$\partial A \cap B^\circ \neq \varnothing$
    if $x = \{A\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 001$ //$A^\circ \cap \partial B \neq \varnothing$
  if $N = \mathring{nbr}$ **then** //*Check if tree received from all neighbors*
    if $\mathring{bnum} \neq 000$ **then**
      become BNDY
    else
      become DONE

BNDY
Spontaneously
  send (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Initiate message to sink*
  become DONE

DONE, IDLE
Receiving (`rprt`, b)
  if $b \vee \mathring{bnum} \neq \mathring{bnum}$ **then** //*Check for new data*
    set $\mathring{bnum} := b \vee \mathring{bnum}$ //*Data aggregation*
    send (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Forward aggregate data*

- The single INIT sink node begins the algorithm as normal by broadcasting a `ping` message before transitioning to state SINK.
- `ping` messages are flooded throughout the network, enabling nodes to identify their parents in the routing tree, and determine whether they are at the boundary of $A \cap B$ (and so required to initiate a `rprt` message).
- Additionally, each node includes in its `ping` message its current knowledge of its bit sequence, $\mathring{bnum}$. In cases where the sink is *outside* $A \cap B$, this information enables nodes in $A^\circ \cap B^\circ$ to determine if they are also one-

Algorithm 3 Four-bit fine resolution topological relations between regions A and B (4bit)

1: Restrictions: reliable communication; connected, undirected graph $G = (V, E)$; $nbr : V \to \mathscr{P}(V)$ where $nbr(v) \mapsto \{v' | \{v, v'\} \in E\}$; $sense : V \to \mathscr{P}(\{A, B\})$
2: State Trans. Sys.: $\langle \{\text{SINK}, \text{INIT}, \text{IDLE}, \text{BNDY}, \text{DONE}\}, \{(\text{INIT}, \text{SINK}), (\text{IDLE}, \text{BNDY}), (\text{IDLE}, \text{DONE})\} \rangle$
3: Initialization: All nodes in state IDLE, except one node in INIT
4: Local variables: $bnum : V \to \mathbb{B}^4$, initialized $\mathring{bnum} := 0000$; $nB\mathring{num} : V \to \mathbb{B}^4$, initialized $nB\mathring{num} := 0000$; $parent : V \to V \cup \{\varnothing\}$, initialized $\mathring{parent} := \varnothing$; visited neighbors N, initialized $N := \varnothing$

INIT
5: *Spontaneously*
6: **broadcast** (`ping`, $\mathring{sense}$, $\mathring{bnum}$) //*Sink initiates algorithm*
7: **become** SINK

SINK
8: *Receiving* (`rprt`, b)
9: **set** $\mathring{bnum} := b \vee \mathring{bnum}$
10: Deduce topological relation between A and B from $\mathring{bnum}$ according to Table 1

IDLE
11: *Receiving* (`ping`, x, nb) from v
12: $N := N \cup \{v\}$ //*Update list of visited nodes*
13: $nB\mathring{num} := nB\mathring{num} \vee nb$ //*Update neighbor's bnum*
14: **if** $x \neq \mathring{sense}$ and $\mathring{sense} = \{A, B\}$ **then**
15: **if** $x = \varnothing$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 1000$ //$\partial A \cap \partial B \neq \varnothing$
16: **if** $x = \{B\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0100$ //$\partial A \cap B^\circ \neq \varnothing$
17: **if** $x = \{A\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0010$ //$A^\circ \cap \partial B \neq \varnothing$
18: **if** $\mathring{parent} = \varnothing$ **then** //*Check for first* `ping` *received*
19: **set** $\mathring{parent} := v$ //*Store tree parent*
20: **broadcast** (`ping`, $\mathring{sense}$, $\mathring{bnum}$) //*Continue building tree*
21: **if** $N = \mathring{nbr}$ **then** //*Check if tree received from all neighbors*
22: **if** $\mathring{sense} = \{A, B\}$ **then**
23: **if** $\mathring{bnum} = 0000$ **then** //$A^\circ \cap B^\circ \neq \varnothing$
24: **if** $nB\mathring{num} \neq 0000$ **then** //*One of this node's neighbor is boundary node*
25: **set** $\mathring{bnum} := \mathring{bnum} \vee 0001$
26: **send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Initiate message to sink*
27: **else**
28: **set** $\mathring{bnum} := \mathring{bnum} \vee 0001$ //*Update local bit number*
29: **if** $\mathring{bnum} \neq 0000$ or $\mathring{bnum} \neq 0001$ **then**
30: **become** BNDY
31: **else**
32: **become** DONE

BNDY
33: *Spontaneously*
34: **send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Initiate message to sink*
35: **become** DONE

DONE, IDLE
36: *Receiving* (`rprt`, b)
37: **if** $b \vee \mathring{bnum} \neq \mathring{bnum}$ **then** //*Check for new data*
38: **set** $\mathring{bnum} := b \vee \mathring{bnum}$ //*Data aggregation*
39: **send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Forward aggregate data*

hop neighbors of the boundary of $A \cap B$. In Algorithm 3 these nodes also initiate `rprt` messages (see Algorithm 3 line 26).
- In cases where the sink is *inside* $A \cap B$, the returning `rprt` messages must be forwarded in any case by nodes in $A^\circ \cap B^\circ$ that are one-hop neighbors of nodes at the boundary of $A \cap B$. Accordingly, these nodes are able to add their knowledge to the `rprt` message (see Algorithm 3 line 28).
- As before, nodes receiving a `rprt` check that the message contains new, unseen information before forwarding further (Algorithm 3, lines 37–39).
- In the final step, the sink node deduces topological relations again based on Table 1.

3.5 Intersection and difference model for complex regions (ID)

All the algorithms already discussed deal only with simple regions homeomorphic to a disk. While this is an important starting point, it is obviously not sufficient to model the variety and complexity of geographic regions (i.e., complex areal objects with multiple disconnected components and holes). Several models of topological relations between complex areal objects have already been discussed in section 2. However, as for simple regions, none of these models can be directly applied in a decentralized spatial algorithm because of the inherent constraints to communication in a geosensor network.

Of the alternatives, the most appropriate for adaptation to a decentralized spatial computing environment is the intersection and difference model [4]. In contrast to the 4-intersection model ($\partial A \cap \partial B = \varnothing$, $\partial A \cap B^\circ = \varnothing$, $A^\circ \cap \partial B = \varnothing$, $A^\circ \cap B^\circ = \varnothing$), the four conditions tested in the intersection and difference model are: $\partial A \cap \partial B = \varnothing$, $A - B = \varnothing$, $B - A = \varnothing$, $A^\circ \cap B^\circ = \varnothing$. Table 2 summarizes the extensions made to the intersection and difference model, distinguishing the eight four-bit sequences taken directly from the model, from the seven four-bit sequences added to the model in order to account for the finite-granularity characteristics of a geosensor network. For instance, if two regions are empty, the description of disjoint could include 0000 (not considered in the intersection and difference model). If two regions have no interiors, the equals relation can include the 1000 bit sequence. Similarly, coarse spatial granularity can lead to a lack of region interiors for the bit-sequences 1100 (cover), 1010 (covered by), 0100 (contain), and 0010 (inside). Finally, 0111 is classed as overlap, but where no single node in $A \cap B$ is a one-hop neighbor of a node outside both A and B (due to limited granularity, as already seen in Fig 5a). The only remaining bit sequence, 0001, is not possible (i.e., non-empty interior intersection, but empty boundary intersections, and empty set differences $A - B$ and $B - A$ cannot occur).

With these adaptations, this extended model can determine the topological relations between complex areal objects with disconnected parts. For example, in Fig. 9 the sink node will infer an overall bit sequence of $0101 \vee 0010 = 0111$, corresponding to an overlap relation (Table 2). By contrast, using the extended 4-intersection model in Algorithm 3 would result in incorrectly infer-

Table 2 Determining the topological relation between complex regions using the intersection and difference model (Algorithm 4). Four-bit numbers represent $\langle \partial A \cap \partial B \neq \varnothing, A - B \neq \varnothing, B - A \neq \varnothing, A^\circ \cap B^\circ \neq \varnothing \rangle$. Bit sequences in bold font indicate additions to the intersection and difference model of [4] to account for limited granularity of geosensor network. Bit sequences not tested in our simulations are indicated using †.

bnum(v) for sink node v $\in V$	Topological relation
0110, **0000**†	A, B disjoint
1110	A, B meet
1111, **0111**	A, B overlap
1001, **1000**	A, B equal
0101, **0100**	A contains B
0011, **0010**	A inside B
1101, **1100**	A covers B
1011, **1010**	A covered by B

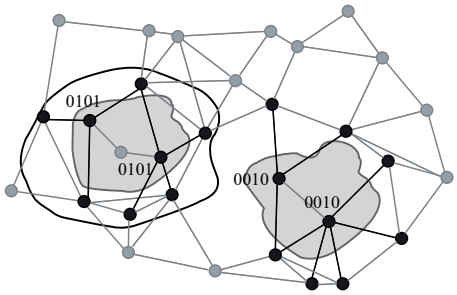


Fig. 9 An overlap relation based on the intersection and difference model between two regions, one with disconnected parts

ring the inside relation between regions, effectively ignoring the disconnected component. The resulting eight topological relations correspond directly to the coarse-granularity grouping presented in [30] of the full 33 topologically distinct relations between complex areal objects. Finally, Algorithm 4 provides the decentralized spatial algorithm corresponding to the bit sequences in Table 2. The algorithm itself adopts the same basic structure and techniques as already encountered in Algorithm 3.

3.6 Summary

All four algorithms are expected to have overall computational complexity O(n), and load balance O(1), since in all cases every node can send at most five messages: one `ping` message, and up to four `rprt` messages (and only three in the case of Algorithm 2).

While every node is required to broadcast a `ping` message, arguably the cost of this operation can be amortized by the cost of network initialization. In establishing an ad hoc network, nodes would in any event be required to broadcast a "hello" message to initiate communication. The information

Algorithm 4 Computing topological relations between complex areal objects A and B, based on the intersection and difference model (ID)

1: Restrictions: reliable communication; connected, undirected graph $G = (V, E)$; $nbr : V \to \mathscr{P}(V)$ where $nbr(v) \mapsto \{v' | \{v, v'\} \in E\}$; $sense : V \to \mathscr{P}(\{A, B\})$
2: State Trans. Sys.: $\langle \{\text{SINK}, \text{INIT}, \text{IDLE}, \text{BNDY}, \text{DONE}\}, \{(\text{INIT}, \text{SINK}), (\text{IDLE}, \text{BNDY}), (\text{IDLE}, \text{DONE})\}\rangle$
3: Initialization: All nodes in state IDLE, except one node in INIT
4: Local variables: $bnum : V \to \mathbb{B}^4$, initialized $\mathring{bnum} := 0000$; $nB\mathring{num} : V \to \mathbb{B}^4$, initialized $nB\mathring{num} := 0000$; $parent : V \to V \cup \{\varnothing\}$, initialized $\mathring{parent} := \varnothing$; visited neighbors N, initialized N $:= \varnothing$;

INIT
5: *Spontaneously*
6: **broadcast** (`ping`, $\mathring{sense}$, $\mathring{bnum}$) //*Sink initiates algorithm*
7: **become** SINK

SINK
8: *Receiving* (`rprt`, b)
9: **set** $\mathring{bnum} := b \vee \mathring{bnum}$
10: Deduce topological relation between A and B from $\mathring{bnum}$ according to Table 2

IDLE
11: *Receiving* (`ping`, x) from v
12: $N := N \cup \{v\}$ //*Update list of visited nodes*
13: **if** $x \neq \mathring{sense}$ and $\mathring{sense} = \{A\}$ **then**
14: **if** $x = \{B\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0100$ //*Node at boundary* $A - B$
15: **if** $x = \varnothing$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0100$ //*Node at boundary* $A - B$
16: **if** $x \neq \mathring{sense}$ and $\mathring{sense} = \{B\}$ **then**
17: **if** $x = \{A\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0010$ //*Node at boundary* $B - A$
18: **if** $x = \varnothing$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0010$ //*Node at boundary* $B - A$
19: **if** $x \neq \mathring{sense}$ and $\mathring{sense} = \{A, B\}$ **then**
20: **if** $x = \{B\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0010$ //$B - A \neq \varnothing$
21: **if** $x = \{A\}$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 0100$ //$A - B \neq \varnothing$
22: **if** $x = \varnothing$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 1000$ //$\partial A \cap \partial B \neq \varnothing$
23: **if** $\mathring{parent} = \varnothing$ **then** //*Check for first* `ping` *received*
24: **set** $\mathring{parent} := v$ //*Store tree parent*
25: **broadcast** (`ping`, $\mathring{sense}$, $\mathring{bnum}$) //*Continue building tree*
26: **if** $N = \mathring{nbr}$ **then** //*Check if tree received from all neighbors*
27: **if** $\mathring{sense} = \{A, B\}$ **then**
28: **if** $\mathring{bnum} = 0000$ **then** //$A^\circ \cap B^\circ \neq \varnothing$
29: **if** $nB\mathring{num} \neq 0000$ **then** //*One of this node's neighbor is boundary node*
30: **set** $\mathring{bnum} := \mathring{bnum} \vee 0001$
31: **send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Initiate message to sink*
32: **else**
33: **set** $\mathring{bnum} := \mathring{bnum} \vee 0001$ //*Update local bit number*
34: **if** $\mathring{bnum} = 0110$ **then set** $\mathring{bnum} := \mathring{bnum} \vee 1000$ //$\partial A \cap \partial B \neq \varnothing$
35: **if** $\mathring{bnum} \neq 0000$ or $\mathring{bnum} \neq 0001$ **then**
36: **become** BNDY
37: **else**
38: **become** DONE

BNDY
39: *Spontaneously*
40: **send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Initiate message to sink*
41: **become** DONE

DONE, IDLE
42: *Receiving* (`rprt`, b)
43: **if** $b \vee \mathring{bnum} \neq \mathring{bnum}$ **then** //*Check for new data*
44: **set** $\mathring{bnum} := b \vee \mathring{bnum}$ //*Data aggregation*
45: **send** (`rprt`, $\mathring{bnum}$) to $\mathring{parent}$ //*Forward aggregate data*

contained in the `ping` message could easily be included in this required network initialization step.

Following this reasoning, the number of `rprt` messages could be proportional to $|V|$ in the worst case for Algorithm 1. However, Algorithms 2–4 require only nodes at the *boundary* of the intersection of regions A and B to initiate `rprt` messages (or their one-hop neighbors in the cases of Algorithms 3 and 4). As a result, substantially fewer nodes are expected to participate in these messages, and the overall communication complexity of `rprt` messages for those algorithms is expected to be $O(n^k)$, $0.5 \leq k < 1$.

These expectations are evaluated experimentally in the following section.

4 Experiments

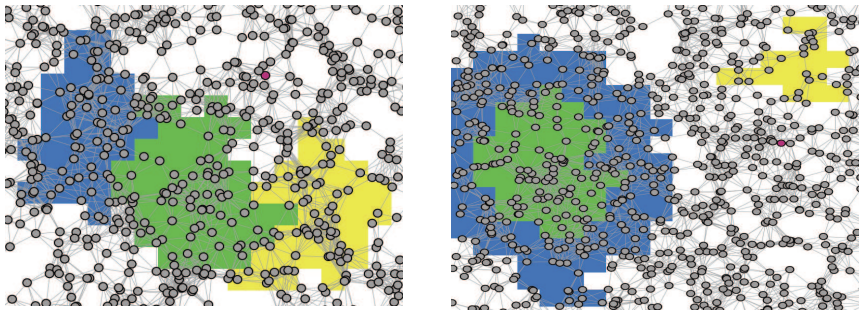
The performance of the four algorithms was evaluated and compared with respect to three criteria: the overall communication complexity, the load balance, and the accuracy of the responses generated by the algorithms.

4.1 Experimental setup

The algorithms mentioned in the previous section were implemented within the agent-based simulation system, NetLogo [37]. NetLogo is particularly well-adapted to these types of experiments for two reasons: it enables algorithms to be implemented using code that is extremely close to the formal specification, reducing impedance mismatch; and it allows the simulation of both the geosensor network and the geographic environment being monitored, a feature absent from many purpose-built sensor network simulation systems.

For each simulation run, a geosensor network with randomized node locations was generated. The network was connected by the unit distance graph (UDG). The UDG models the physical network structure, where nodes closer than some communication distance c can engage in direct one-hop communication. In order to ensure comparability across simulations, the level of network connectivity was kept constant regardless of the network size. In practice, this means that as the number of nodes in the simulation area is doubled (increasing node density), the communication distance c is reduced by a factor of $\sqrt{2}$ accordingly.

Further, a randomly generated pair of regions was grown using a randomized variant of a dilation operation from image processing. The dilation procedure was constrained to ensure that the experimenter can control the topological relation of the regions generated. Thus, the spatial extent of a geosensor network can cover the whole boundaries of generated regions. This can guarantee that all regions will be monitored. Furthermore, a boundary must necessarily lie somewhere in-between a node that detects a region, and a neighbor that lies outside a region (as shown in Fig. 2). This has now become one of the standard definitions of boundary used across research into geosensor



(a) (b)

Fig. 10 Example of randomly generated overlapping regions monitored by randomly generated network: a. simple regions; b. complex areal object (in color blue indicates region $A - B$, yellow indicates region $B - A$, and green indicates $A \cap B$.

Table 3 Results of power regression analysis ($y = ax^b$) for responsive curves in Fig.11

Algorithms	Factor a	Power b	R^2
Basic (1)	0.2871	0.9176	0.98388
3bit (2)	0.642	0.7443	0.96933
4bit (3)	0.6767	0.7602	0.95696
ID (4)	1.7501	0.7283	0.96553

networks [7, 13, 18, 33]. Fig. 10 shows two examples of randomly generated regions that overlap (simple region in Fig. 10a and complex areal object with a disconnected part in Fig. 10b) as well as the UDG connecting nodes.

4.2 Overall scalability

A series of experiments to investigate overall scalability was conducted across seven different network sizes (250, 500, 1000, 2000, 4000, 8000, and 16,000 nodes), for each of the 8 topological relations and four different algorithms, with 10 randomized replications at each level (total $7 \times 8 \times 4 \times 10 = 2240$ simulation runs). For each run the overall number of `ping` and `rprt` messages generated was measured.

As expected, the `ping` messages generate exactly $|V|$ messages in all cases. However, as already argued, `ping` messages could be regarded as part of the required ad hoc network initialization costs. Consequently, the more salient feature of the algorithm from the perspective of efficiency is the `rprt` messages generated.

Fig. 11 shows the total number of `rprt` messages generated for the overlap relation only, for each algorithm and averaged over 10 randomized simulations. The results of a regression analysis on the same data is shown in Table 3. All the regression curves achieved relatively good fit, indicated by R^2 values ranging from 0.95 to 0.98.

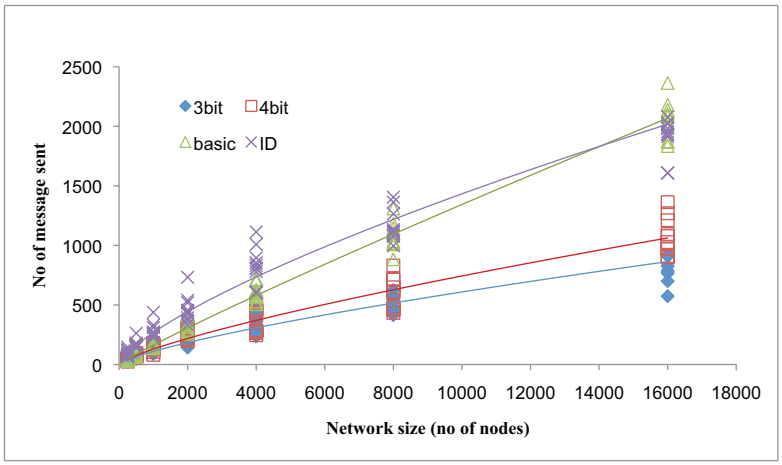


Fig. 11 Overall scalability for `rprt` messages for the overlap relation

The results can be interpreted to indicate that Algorithms 2 showed comparable performance to Algorithm 3, with Algorithm 4 arguably approximately the same order ($O(n^{0.73})$) but larger constant factor (1.75 as compared with *approx* 0.6). Despite a substantially lower factor ($\approx$0.3) Algorithm 1 arguably has the worst scalability given its substantially higher order ($O(n^{0.92})$). This broadly agrees with our expectation, where Algorithm 2 generates only messages at the boundary of the intersection of A and B, and at most three `rprt` messages for any node; Algorithm 3 adds up to four `rprt` messages for any node, and `rprt` messages from one-hop neighbors of the boundary of the intersection of A and B; Algorithm 4 further generates messages from the boundaries of A and B; and Algorithm 1 generates messages from both the boundary and interior of the intersection of A and B.

Further investigations investigated if the results for the different algorithms were significantly different, or if differences might have arisen by chance. Accordingly, the following hypotheses were formulated for all pairs of algorithms x and y:

H_0: $\mu_x - \mu_y = 0$ for all network sizes
H_1: $\mu_x - \mu_y \neq 0$ for some network sizes
H_2: $\mu_x - \mu_y \neq 0$ for all network sizes

Using a t-test for the difference between two means (i.e. the means of the samples for each pair of algorithms μ_x and μ_y), the null hypothesis H_0 was rejected in all cases at the 1% level. Further, the hypothesis H_1 was rejected in favor of the stronger H_2 at the 5% level when comparing Algorithms 1 and 3, 2 and 3, and 2 and 4. The majority of these exceptions relate to the smaller granularities (where the total number of messages generated is in any event smaller). But the general pattern is clear: there is strong evidence that the Algorithms do give rise to significantly different scalability.

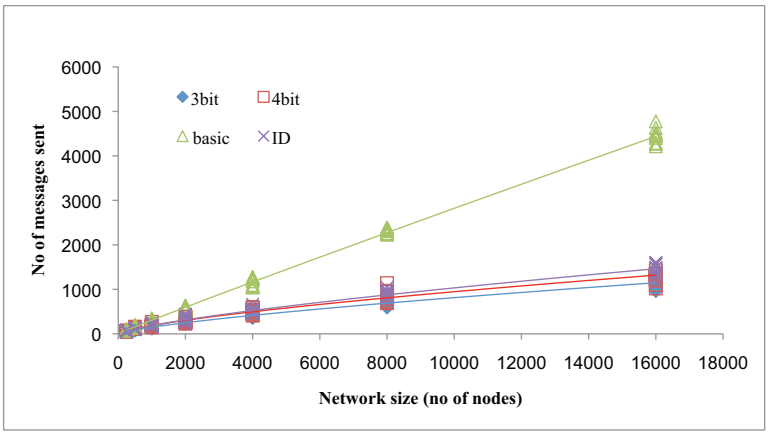


Fig. 12 Overall scalability of `rprt` message for equals relation

Table 4 Resutls of power regression analysis ($y = ax^b$) for responsive curves in Fig.12

Algorithms	Factor a	Power b	R^2
Basic (1)	0.3905	0.9647	0.99585
3bit (2)	0.9395	0.7343	0.98684
4bit (3)	1.4329	0.705	0.97691
ID (4)	1.175	0.7359	0.97785

The other seven topological relations all follow similar patterns, with the exception of the disjoint relation (which generates no `rprt` messages in Algorithms 1, 2, and 3). The main differences variability across topological relations can be attributed to differences in the size of the intersection between A and B. For example, assuming constant region sizes, the size of the intersection between A and B must be larger for the equals relation than for the overlap relation. To illustrate, Fig. 12 and Table 4 show the results of the same experiments for the equals relation. In this case, the null hypothesis H_0 was not rejected for Algorithm 3 and Algorithm 4 at the 1% level, indicating no significant difference between the performance of these algorithms for this specific topological relation.

4.3 Load balance

Load balance is a vital characteristic of any decentralized algorithm, arguably more so than overall communication complexity. Uneven load balance will consume energy more rapidly at some nodes. In practice, that can mean some nodes deplete their energy resource and die more rapidly, potentially leading to holes in network coverage and even disconnected networks.

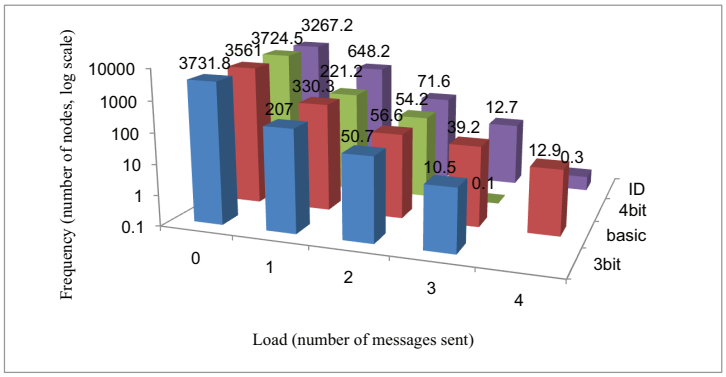


Fig. 13 Load balance for rprt message for overlap relations (averaged over 10 networks of 4000 nodes)

In addition to overall messages sent, the experiments described above also recorded the number of messages sent on a per-node basis. The load histogram in Fig. 13 shows for the overlap relation the number of `rprt` messages transmitted by individual nodes (averaged over the 10 repetitions) against the frequency of nodes transmitting that number of messages. As expected, all algorithms transmitted at most four `rprt` messages (or three in the case of Algorithm 2). Furthermore, the basic algorithm (Algorithm 1) exhibits a substantially larger number of nodes with the largest loads (four messages). Again, comparable results were obtained for all the topological relations.

4.4 Veracity of algorithms

In addition to scalability, the experimental investigation evaluated the veracity of the algorithms, in terms of the accuracy of the detected topological relation. For example, Table 5 shows the misclassification for 100 simulation runs of Algorithm 2 with first 500 nodes and then 4000 nodes. The rows of Table 5 correspond to the actual topological relation, whereas columns correspond to the detected topological relation. Because Algorithm 2 only detects five (coarse granularity) topological relations, the matrix has only five columns, but 8 all possible topological relations were tested (i.e., both meet and overlap at fine topological granularity are detected as overlap at coarse granularity; both contains and covers at fine granularity map to contains at coarse granularity; and inside and covered by both map to inside).

Results for both 500 and 4000 nodes are given together in Table 5. For example, an entry of “91/100” in the M/O cell indicates that 91 out of 100 experimental configurations that were meet were correctly detected as overlap with a network of 500 nodes, whereas all 100 were correctly detected as overlap with a network of 4000 nodes. All the algorithms tested exhibited similar

Table 5 Misclassification matrix (rows are actual relation, columns detected relation) resulting from Algorithm 2. Figures are quoted for network sizes of 500 nodes/4000 nodes. Note: D=disjoint; M=meet; O=overlap; C=contains; I=inside; V=covers; B=covered by; E=equals

	D	O	C	I	E	Total
D	100/100	0/0	0/0	0/0	0/0	100/100
M	0/0	91/100	6/0	3/0	0/0	100/100
O	0/0	100/100	0/0	0/0	0/0	100/100
C	0/0	0/0	100/100	0/0	0/0	100/100
I	0/0	0/0	0/0	100/100	0/0	100/100
V	0/0	0/0	100/100	0/0	0/0	100/100
B	0/0	0/0	0/0	100/100	0/0	100/100
E	0/0	0/0	0/0	0/0	100/100	100/100
Total	100/100	191/200	206/200	203/200	100/100	

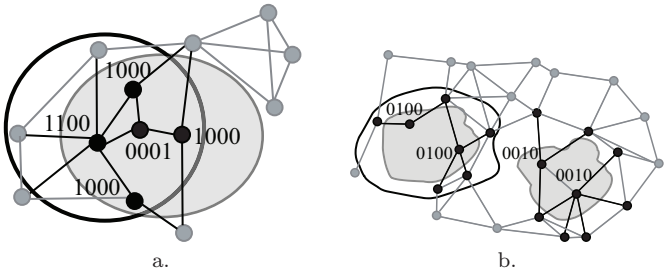
Table 6 Misclassification matrix (rows are actual relation, columns detected relation) resulting from Algorithm 3. Figures are quoted for network sizes of 500 nodes/4000 nodes. Note: D=disjoint; M=meet; O=overlap; C=contains; I=inside; V=covers; B=covered by; E=equals

	D	M	O	C	I	V	B	E	Total
D	100/100	0/0	0/0	0/0	0/0	0/0	0/0	0/0	100/100
M	0/0	98/81	0/19	1/0	1/0	0/0	0/0	0/0	100/100
O	0/0	0/0	100/100	0/0	0/0	0/0	0/0	0/0	100/100
C	0/0	0/0	0/0	97/100	0/0	3/0	0/0	0/0	100/100
I	0/0	0/0	0/0	0/0	97/100	0/0	3/0	0/0	100/100
V	0/0	0/0	0/0	0/0	0/0	100/100	0/0	0/0	100/100
B	0/0	1/0	0/0	0/0	0/0	0/0	100/100	0/0	101/100
E	0/0	0/0	0/0	0/0	0/0	0/0	0/0	100/100	100/100
Total	100/100	99/81	100/119	98/100	98/100	103/100	103/100	100/100	

increases in accuracy with finer spatial granularity (i.e., larger, more dense networks), such that at 4000 nodes Table 5 indicates Algorithm 2 was operating at perfect accuracy.

The picture was slightly more complicated in the case of the fine topological granularity algorithms (Algorithms 1, 3, and 4). For example, Table 6 broadly shows the same pattern of increasing topological accuracy with finer spatial granularity (e.g., at 500 nodes, three contains relations were misclassified as covers; at 4000 nodes all contains relations were correctly classified). However, increasing the network size actually leads to *more* instances of the meet relation being misclassified (98 out of 100 correctly classified at network size of 500 nodes; 81 out of 100 correctly classified at 4000 nodes). This pattern was repeated for all three algorithms that operated at fine topological granularity (i.e., Algorithms 1, 3, and 4).

This issue arises because meet is not well-defined in the discrete case. Because of the finite granularity of the geosensor networks, the boundary of the region can never be detected directly, but rather as a node that is strictly inside the region with a one-hop neighbor that is outside the region. Thus, a *true* meet in the continuous (infinite granularity) space would always be detected



a. b.

Fig. 14 Misclassifications of topological relations caused by the limited spatial granularity, for example: a. where no node in $B-A$ results in overlap misclassified as cover; and b. where no node in $A^\circ \cap B^\circ$ results in overlap misclassified as disjoint (for complex areal object B with disconnected components).

as disjoint in the discrete, finite granularity case; instead regions must *slightly* overlap in the continuous case in order to be detected as meet in the discrete case. Note that the same is not true of the other two boundary-critical relations introduced by the finer granularity topological models, covers and covered by. For example, a region that covers another in the continuous case may also be detected as covers in the discrete case (although the converse is not true—a region that is detected as covering another in the discrete space does not necessarily do so in the continuous space). Moving from low to higher densities of nodes, small enough overlaps to be detected as a meet relation become more likely to be detected as an overlap at finer spatial granularities. We shall return to this issue in the discussion and conclusions.

Investigations of all the other topological misclassifications revealed two further distinct causes of misclassifications. In the first case, the limited spatial granularity of the network and random node locations could sometimes mean that no node happened to be located in some critical region component. For example, Fig. 14 provides two concrete examples where the chance absence of any node in $B - A$ (Fig. 14a) or in $A^\circ \cap B^\circ$ (Fig. 14b) at coarse spatial granularity leads to misclassifications of the actual topological relation in the continuous space.

In the second case, even when there existed nodes in key region components, topological misclassifications could occur either by the absence of direct network connection (e.g., caused by network holes and uneven distribution of nodes, Fig. 15a); or where network direct connections spanned two close, but not coincident boundaries (e.g., Fig. 15b). These were still in effect granularity issues, but network granularity rather than spatial granularity issues (i.e., the problems could potentially have been averted if different network connections had arisen).

In summary, while all the algorithms performed at high levels of accuracy, especially for larger and more dense network sizes, the three main causes of misclassifications were:

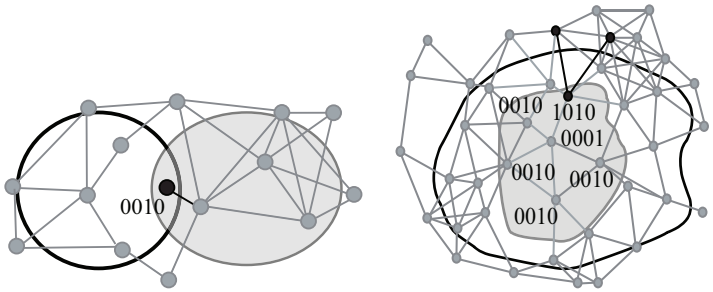


Fig. 15 Misclassifications of topological relations caused by the network granularity, for example: a. where a meet is misclassified as contain; and b. where contain is misclassified as cover

- the weak definition of meet for discrete, finite granularity observations of regions, requiring that regions in the continuous space must in fact slightly overlap to be detected as meet;
- pure spatial granularity issues, where nodes did not happen to be located in critical components of the regions; and
- network granularity issues, where adverse connectivity close to boundaries happened to obscure the true topological relation.

5 Discussion and conclusions

This paper has demonstrated how spatial queries about topological relation between simple and complex regions can be satisfied using decentralized algorithms. The approach combines efficient data aggregation with spatial filtering, targeting communication at those nodes at or near the boundary of monitored regions. Significantly, the algorithms do not depend on quantitative coordinate information, only on relative neighborhood information. Because qualitative neighborhood information is expected to be available at the most basic level to any geosensor network, the algorithm is well-suited to highly resource-constrained networks, with no access to GPS or other positioning systems. Thus, even though we can use computational geometry to determine topological relations between regions, this research does not require the whole nodes to have coordinate information.

All four algorithms were highly scalable, with overall $O(n)$ communication complexity and optimal load balance $O(1)$, verified through experimental simulation. Indeed, if the $O(n)$ initialization step for each algorithm is treated as an unavoidable component of ad hoc network establishment, then for Algorithms 2–4 the overall computational complexity reduced to $O(n^k)$, where $0.5 \leq k < 1$ (and in our experiments $0.7 < k < 0.8$). Algorithm 2 was consistently amongst the most efficient in our experiments, but at the cost of reduced topological granularity, yielding only five (rather than 8) distinct topological relations. It is arguable that distinguishing between the three additional topo-

logical relations (meet versus overlap, covers verses contains, covered by versus inside) is not of high interest in many applications. However, the detection of fine granularity topological relations between simple or even complex regions (Algorithms 3 and 4) was not significantly less scalable that the detection of coarse granularity topological relations (Algorithm 2) in many experiments.

When considering the veracity of the algorithms, all the algorithms performed at high levels of accuracy, identifying the correct topological relation in the overwhelming majority of cases. However, granularity effects did cause misclassifications, in particular with smaller, less dense networks; regions that were small or had close boundaries when compared with the network density; and with the meet relation, which is difficult to model adequately in discrete spaces. A key issue for future work is to investigate more appropriate models of topological relations between granular spatial regions monitored by a geosensor network. This question is related to past work, for example in digital topology (e.g., [24, 38]). However, this past work relies substantially on the regularity of neighborhoods in the raster images towards which this work was targeted. The highly irregular neighborhoods in a geosensor network require more generalized descriptions of discrete topology.

Our approach does only address topological relations between two regions. It is, however, not especially challenging to extend our algorithms to determine the topological relations between multiple regions, for example by combining the pairwise approach used in this paper. Further work might also implement and test the algorithms in a real sensor network with relative ease, although at considerable financial cost with today's technology. A basic assumption behind this work is that while today's geosensor networks are complex, expensive, and typically numbered in hundreds of nodes, tomorrow's networks will be based on cheaper technology with thousands or even millions of nodes, providing much more spatial detail about the environment. This progress towards cheaper, more accessible technology is very evident even over the past 5 years.

Finally, the algorithms in this paper are specifically designed to query the static topological relations between regions, suitable for infrequent, one-off (snapshot) queries. As the frequency of queries increases, so the efficiency of any snapshot approach decreases. In cases where high frequency or long-running queries are required, event-based approaches that monitor topological *changes* (merging and splitting of regions) rather than snapshot-based approaches are likely to become appropriate (such as [13]). However, we argue that these two perspectives are complementary: to satisfy the requirements of a range of applications, algorithms for both snapshot and long-running queries are needed. Further, long running queries typically need to be initialized using a snapshot query, like those investigated in this paper, in order to correctly infer the changes that are occurring.

Acknowledgements This work was supported under the Australian Research Council (ARC) Future Fellowship funding scheme (grant number FT0990531) and the ARC Discovery Project scheme (grant number DP120103758).

References

1. Bredon GE (1993) Topology and Geometry. Springer-Verlag, New York
2. Clementini E, Felice PD, Califano G (1995) Composite regions in topological queries. Information Systems 20(7):579 – 594
3. Cohn AG, Bennett B, Gooday J, Gotts NM (1997) Qualitative spatial representation and reasoning with the region connection calculus. GeoInformatica 1:275–316, URL http://dx.doi.org/10.1023/A:1009712514511
4. Deng M, Cheng T, Chen X, Li Z (2007) Multi-level topological relations between spatial regions based upon topological invariants. Geoinformatica 11(2):239–267
5. Duckham M (2012, in press) Decentralized Spatial Computing: Foundations of Geosensor Networks. Springer: Berlin.
6. Duckham M, Jeong MH, Li S, Renz J (2010) Decentralized querying of topological relations between regions without using localization. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, New York, NY, USA, GIS '10, pp 414–417
7. Duckham M, Nussbaum D, Sack JR, Santoro N (2011) Efficient, decentralized computation of the topology of spatial regions. IEEE Transactions on Computers 60:1100–1113
8. Egenhofer MJ, Fransoza RD (1991) Point-set topological spatial relations. International Journal of Geographical Information Systems 5(2):161–174
9. Egenhofer MJ, Herring J (1992) Categorizing binary topological relationships between regions, lines, and points in geographic databases. Tech. rep., Department of Surveying Engineering, University of Maine, Orono, ME
10. Egenhofer MJ, Clementini E, Di Felice P (1994) Topological relations between regions with holes. International Journal of Geographical Information Systems 8(2):129–144
11. Farah C, Zhong C, Worboys M, Nittel S (2008) Detecting topological change using a wireless sensor network. In: Cova T, Beard K, Goodchild M, Frank A (eds) GIScience 2008, Springer, Berlin, Lecture Notes in Computer Science, vol 5266, pp 55–69
12. Greenwald MB, Khanna S (2004) Power-conserving computation of order-statistics over sensor networks. In: Proc 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), ACM, New York, pp 275–285
13. Guan LJ, Duckham M (2011) Decentralized reasoning about gradual changes of topological relationships between continuously evolving regions. In: Egenhofer MJ, Giudice NA, Moratz R, Worboys MF (eds) Conference on Spatial Information Theory (COSIT '11), Springer, Berlin, Lecture Notes in Computer Science, vol 6899, pp 126–147
14. Hatcher A (2005) Notes on introductory point-set topology. URL http://www.math.cornell.edu/%7Ehatcher/Top/TopNotes.pdf

15. Jiang J, Worboys M (2008) Detecting basic topological changes in sensor networks by local aggregation. In: Proc. 16th ACM International Conference on Advances in Geographic Information Systems (ACMGIS), ACM, New York, pp 1–10
16. Khan A, Schneider M (2010) Topological reasoning between complex regions in databases with frequent updates. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, New York, NY, USA, GIS '10, pp 380–389
17. Krishnamachari B, Estrin D, Wicker SB (2002) The impact of data aggregation in wireless sensor networks. In: Proc. 22nd International Conference on Distributed Computing Systems (ICDCS), IEEE, pp 575–578
18. Lian J, Chen L, abd Yunhao Liu KN, Agnew GB (2007) Gradient boundary detection for time series snapshot construction in sensor networks. IEEE Transactions on Parallel and Distributed Systems 18(10):1462–1475
19. Madden S, Franklin MJ, Hellerstein JM, Hong W (2002) TAG: A tiny aggregation service for ad-hoc sensor networks. In: Proc. 5th Symposium on Operating System Design and Implementation (OSDI), pp 131–146
20. Mandelbrot BB (1977) Fractals, Form, Chance And Dimnsion. San Francisco
21. Nguyen V, Parent C, Spaccapietra S (1997) Complex regions in topological queries. In: Hirtle S, Frank A (eds) Spatial Information Theory A Theoretical Basis for GIS, Lecture Notes in Computer Science, vol 1329, Springer Berlin / Heidelberg, pp 175–192
22. Nittel S (2009) A survey of geosensor networks: Advances in dynamic environmental monitoring. Sensors 9(7):5664–5678
23. Randell DA, Cui Z, Cohn AG (1992) A spatial logic based on regions and connection. In: Nebel B, Swartout W, Rich C (eds) Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference, pp 156–176
24. Rosenfeld A (1979) Digital topology. The American Mathematical Monthly 86(8):621–630
25. Rosso R, Bacchi B, Barbera PL (1991) Fractal relation of mainstream length to catchment area in river networks. Water Resources Research 27(4):381–387
26. Sadeq M, Duckham M (2008) Effect of neighborhood on in-network processing in sensor networks. In: Cova T, Beard K, Goodchild M, Frank A (eds) GIScience 2008, Springer, Berlin, Lecture Notes in Computer Science, vol 5266, pp 133–150
27. Sadeq MJ, Duckham M (2009) Decentralized area computation for spatial regions. In: 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp 432–435
28. Santoro N (2007) Design and analysis of distributed algorithms. New Jersey: Wiley
29. Sarkar R, Zhu X, Gao J, Guibas LJ, Mitchell JSB (2008) Iso-contour queries and gradient descent with guaranteed delivery in sensor networks. In: Proc. 27th IEEE Conference on Computer Communications (INFO-

COM), pp 960–967
30. Schneider M, Behr T (2006) Topological relationships between complex spatial objects. ACM Transactions on Database Systems 31(1):39–81
31. Sharifzadeh M, Shahabi C (2005) Utilizing Voronoi cells of location data streams for accurate computation of aggregate functions in sensor networks. Geoinformatica 10(1):9–36
32. Shen G (2002) Fractal dimension and fractal growth of urbanized areas. International Journal of Geographical Information Science 16(5):419 — 437
33. Shi M, Winter S (2010) Detecting change in snapshot sequences. In: Fabrikant S, Reichenbacher T, van Kreveld M, Schlieder C (eds) Geographic Information Science, Lecture Notes in Computer Science, vol 6292, Springer Berlin / Heidelberg, pp 219–233
34. Shrivastava N, Buragohain C, Agrawal D, Suri S (2004) Medians and beyond: New aggregation techniques for sensor networks. In: Proc. 2nd International Conference on Embedded Networked Sensor Systems (SenSys), ACM, New York, pp 239–249
35. Skraba P, Fang Q, Nguyen A, Guibas L (2006) Sweeps over wireless sensor networks. In: 5th Int'l Conference on Information Processing in Sensor Networks (IPSN), pp 143–151
36. Tarboton DG, Bras RL, Rodriguez-Iturbe I (1988) The fractal nature of river networks. Water Resources Research 24(8):1317–1322
37. Wilensky U (1999) Netlogo. URL `http://ccl.northwestern.edu/netlogo/`
38. Winter S (1995) Topological relations between discrete regions. In: Egenhofer M, Herring J (eds) Advances in Spatial Databases, Springer, Berlin, Lecture Notes in Computer Science, vol 951, pp 310—327
39. Worboys MF, Duckham M (2006) Monitoring qualitative spatiotemporal change for geosensor networks. International Journal of Geographical Information Science 20(10):1087–1108
40. Zheng R, Barton R (2007) Toward optimal data aggregation in random wireless sensor networks. In: Proc. 26th IEEE International Conference on Computer Communications (INFOCOM), IEEE, Washington, DC, pp 249–257