Decentralized querying of topological relations between regions without using localization

Matt Duckham
University of Melbourne
Victoria 3010, Australia
mduckham@unimelb.edu.au

Myeong Hun Jeong
University of Melbourne
Victoria 3010, Australia
mh.jeong@pgrad.unimelb.edu.au

Sanjiang Li
University of Technology Sydney
NSW 2007, Australia
Sanjiang.Li@uts.edu.au

Jochen Renz
Australian National University
Canberra, ACT 0200, Australia
jochen.renz@anu.edu.au

ABSTRACT

This paper proposes an efficient, decentralized algorithm for determining the topological relationship between two regions monitored by a geosensor network. Many centralized algorithms already exist for this purpose (used for example in spatial databases). However, these algorithms are not suited to decentralized spatial computing environments, like geosensor networks, which must operate without global knowledge of the system state and without centralized control. Unlike many existing decentralized spatial algorithms, the proposed algorithm is also able to operate in the absence of information about a node's coordinate location. This makes the algorithm suitable for applications of geosensor networks where GPS or other positioning systems are unavailable or unreliable. The algorithm approach is founded on the well-known 4-intersection model, using in-network data aggregation and spatial filtering (involving nodes only at some region boundaries). This ensures only a relatively small proportion of the network is involved in computation, thus increasing efficiency. Our analysis shows that while the overall communication complexity of the algorithm is $O(n)$, the load balancing is optimal leading to a constant $O(1)$ communication complexity for individual nodes. This expectation is confirmed with empirical investigation using simulation, which demonstrates the practical efficiency of the algorithm.

Categories and Subject Descriptors

2.4 [**Distributed Systems**]: Distributed applications

Keywords

decentralized spatial computing, geosensor networks, qualitative spatial reasoning, data aggregation, 4-intersection

1. INTRODUCTION

The primary contribution of this paper is to propose and analyze an efficient algorithm for determining the topological relationship between two regions monitored by a geosensor network (a wireless networks of miniaturized sensor-enabled computers monitoring phenomena in geographic space [1]). The unique resource constraints imposed by technologies like geosensor networks mean that traditional, centralized approaches to spatial computation are inefficient and not scalable. In particular, technologies like geosensor networks typically impose high costs on communicating data between nodes (e.g., due to limited node energy), compared with relatively low costs of processing data at the node itself (necessitating so-called *in-network* processing) (cf. [2]).

Consequently, our algorithm is *decentralized*: nodes partially process data in the network without any centralized control. The nodes use only knowledge of their local environment and spatially nearby neighbors, with no system component ever possessing global knowledge of the network state [3]. The algorithm ensures efficiency by using the combination of two decentralized computing strategies. The first strategy is explicitly spatial: restricting communication to the *boundaries* of the regions in question, termed here *spatial filtering*. This strategy substantially reduces the number of nodes that must actively communicate in the algorithm, so improving efficiency. The second strategy is more familiar in distributed computing: *data aggregation*. By ensuring nodes record any information they overhear, the algorithm avoids any information being retransmitted.

When compared with many other decentralized spatial computing algorithms, an important contribution of our algorithm is that it does not rely on quantitative spatial information (such as coordinate location, distance, or direction). The only spatial information required by nodes is their communication neighborhood (i.e., knowledge of which nearby nodes are in direct one-hop radio communication). This information is expected to be available in any wireless sensor network. As a result, the algorithm is suitable for applications where quantitative spatial information is unavailable, such as where power constraints restrict the use of localization systems (like GPS, or direction- or range-finding); where the environment is not favorable for localization (such as using GPS in underwater environments or in dense vegetation); or simply where node hardware does not include localization capabilities.

In the remainder of the short paper, we briefly review our algorithm's characteristics in the context of related work (section 2). Then we present the algorithm itself, and an analysis of its computational characteristics (section 3). Section 4 summarizes an ongoing evaluation of the algorithm, using simulation. Finally, section 5 concludes the paper with a discussion of the limitations of the approach, and future work.

ACMGIS '10 November 2–5, 2010. San Jose, CA, USA

2. BACKGROUND

Research into the application of decentralized algorithms to wireless sensor networks is already well advanced. For example, data aggregation is a fundamental technique used in decentralized systems to eliminate redundancy in transmitted data, reducing the overall communication overheads (and so the energy budget) required by an algorithm [4]. A range of existing algorithms make use of data aggregation (e.g., [5, 6, 7]), and the algorithm proposed in this paper also relies, in part, on data aggregation.

While techniques like data aggregation assist with some simple spatial queries, more sophisticated techniques are also required to efficiently satisfy complex spatial queries. This paper is specifically concerned with spatial 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 sensed directly by nodes in the network (such as the presence or absence of monitored pollutant) or may be derived from thresholding a continuous fields (e.g., temperatures above 30°C) [8]. The approach taken in this paper is to use spatial filtering to improve efficiency, focusing communication on key nodes at boundary of the region.

Thus, the primary contributions of this work are in the development of a new and efficient algorithm for determining the static topological relation of two spatial regions without recourse to coordinate location information.

3. ALGORITHM

Algorithm 1 presents our algorithm, adopting the distributed algorithm presentation style of Santoro [9]. In brief, this approach defines for every node a number of *states* (in upper case letters). In each state, nodes can respond to different *events* (in italics), specifically receiving a message (*Receiving* keyword) or spontaneous events, such as when a node initializes or first transitions to a new state (*Spontaneously* keyword). The responses to each event are specified as atomic *actions*, which are procedures comprising a finite number of steps that the node can complete without interruption (i.e., they require no interaction with other nodes and once started are completed before any further events can be processed by that node).

3.1 Preliminaries

Following previous work, we assume:

- 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. [10]).
- node sensors capable of determining whether a node detects a region A and/or B, modeled as a function *sense* : $V \rightarrow \mathcal{P}(\{A, B\})$. After [11], regions are assumed to be homeomorphic to a disk (no holes or disconnected parts).
- a sink node, responsible for initiating the query and collating the partially processed responses from targeted nodes in the network.

These assumptions are listed as restrictions to Algorithm 1 in line 1. However, individual nodes only have *local* knowledge of 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')). To enforce and highlight the local knowledge to a node, we use the overdot notation $\mathring{sense}$ (termed "local" or "my" *sense*) inside the algorithm 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*$(\circ)$).

3.2 Algorithm

The algorithm can be explained informally by decomposing it into 7 distinct steps:

1. The single node in state INIT initializes the algorithm by broadcasting a `ping` message specifying the regions it can sense locally (lines 7–8).

2. The first time an IDLE node receives a `ping` message, it stores the identity of the sending node (line 13), before rebroadcasting the `ping` message along with its own sensed value (line 14). In this way, a routing tree is constructed for returning data to the sink node at the same time as disseminating the query throughout the network.

3. Nodes that lie in the intersection of A and B (if any) can deduce whether they are at the *boundary* of region A and/or B by comparing their local sensed value with those of their one-hop neighbors. This information is stored as a 4-bit number, *bnum*$(v) \in \mathbb{B}^4$ for a node v. The largest bit indicates whether a node detects an intersection between the boundary of A and the boundary of B (i.e., $\partial A \cap \partial B \neq \varnothing$); the next largest bit indicates whether it detects an intersection between the boundary of A and the interior of B (i.e., $\partial A \cap B^\circ \neq \varnothing$); the third largest bit whether that node detects an intersection between the boundary of B and the interior of A (i.e, $A^\circ \cap \partial B \neq \varnothing$). The smallest bit indicates for nodes at the boundary of both A and B, whether the node has any one-hop neighbors that are in A or B only. This bit enables a node to deduce locally instances where the topological relation cannot be equal. This piece of information is important, since the topological relations meet and equals are identical with respect to the other three bits. Figure 1 gives a simplified example illustrating the four different node states.

4. When nodes in the IDLE state have received `ping` messages from all their network neighbors, they transition either to the BNDY state if they lie at the boundary of the intersection between A and B (line 24) or to the DONE state otherwise (line 26).

5. Nodes that transition to the BNDY state are then able to collate the information gathered to this point so that they store one of four possible 4-bit numbers for *bnum*: 0010 if they are at the boundary of B only; 0100 if they are at the boundary of A only; 1000 if they are at the boundary of A and B, and have no neighbors in A or B only; 1001 if they are at the boundary of A and B, and have neighbors in A or B only (lines 28–29). Note that these four states are MEPD for boundary nodes in the intersection of A and B. These nodes then initiate a report message routed back to the sink, containing their local knowledge of their boundary state (i.e., $\mathring{bnum}$), before also transitioning to state DONE (lines 30–31).

6. Nodes in an DONE state receiving a `rprt` message compare the information contained in the message with their exiting knowledge (either determined directly from their boundary state, or overheard from previously forwarded messages). Only messages that contain new information, not previously known to the node, are forwarded to the sink (lines 33–35). Nodes in a INIT or BNDY state that receive an `rprt` message defer responding to that event, by storing the information in

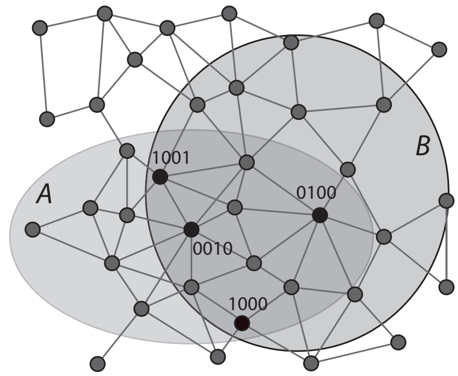

Figure 1: Example of node states and 4-bit encoding

that message and processing only after they have transitioned to DONE (line 37).

7. In the final step, the sink node uses a logical (bitwise) OR operation to compose all the messages received (line 10). Following directly from Egenhofer and Franzosa's 4-intersection model [11], the topological relations that can be deduced from these messages are given in table 1.

$\bigvee R$	Topological relation
0001, 0011, 0101, 0111	Not possible
0000	A, B disjoint
0010	A contains B
0100	B contains A
1000	A, B equal
1001	A, B meet
0110, 1110, 1111	A, B overlap
1010, 1011	A covers B
1100, 1101	B covers A

Table 1: Determining the topological relation between regions (see Algorithm 1, line 29), for set $R \subset \mathbb{B}^3$ of `rprt` message payloads received at sink. 1000 indicates $\partial A \cap \partial B \neq \varnothing$; 0100 indicates $\partial A \cap B^\circ \neq \varnothing$; 0010 indicates $A^\circ \cap \partial B \neq \varnothing$; 0001 indicates $A \neq B$.

3.3 Computational analysis

As communication is the most resource-intensive operation in a sensor network, computational efficiency is typically measured in terms of communication complexity. The communication complexity of the `ping` message is $\Omega(|V|)$ messages sent (every node sends exactly one message). However, initializing any sensor node requires the broadcast of a handshake or "hello" message simply to join the network. Thus, we argue that cost of the `ping` messages can potentially be amortized by the cost of network initialization.

The `rprt` messages are initiated at the boundaries only, following a linear path to the sink, using data aggregation wherever possible (see Algorithm 1, line 33–35). Because the initialization stages of the algorithm structure the paths to the sink as a tree, the expected communication complexity of the `rprt` messages is $O(\log|V|)$. Most importantly, only nodes at the boundary of the intersection between A and B initiate `rprt` messages. While the number of nodes *inside* the intersection of A and B may, in the worst case, scale $O(|V|)$, the number of nodes at the boundary

Algorithm 1 Topological relations between regions A and B

1: Restrictions: $G = (V, E)$, *sense* : $V \to \mathcal{P}(\{A, B\})$
2: States: {SINK, INIT, IDLE, BNDY, DONE}
3: Transitions: {(INIT, SINK), (IDLE, BNDY), (IDLE, DONE), (BNDY, DONE)}
4: Initialization: All nodes in state IDLE, except one node in INIT
5: Local variables: *bnum* : $V \to \mathbb{B}^4$, initialized *bnum* $\mapsto$ 0000; *parent* : $V \to V \cup \{\varnothing\}$, initialized *parent* $\mapsto \varnothing$

INIT
6: *Spontaneously*
7: **broadcast** (`ping`, *sense*) --*Sink initiates algorithm*
8: **become** SINK

SINK
9: *Receiving* (`rprt`, b)
10: Deduce topological relationship between A and B --*See table 1*

IDLE
11: *Receiving* (`ping`, x) from v
12: **if** *parent* $= \varnothing$ **then** --*Check for first `ping` received*
13: **set** *parent* $\mapsto v$ --*Store tree parent*
14: **broadcast** (`ping`, *sense*) --*Continue building tree*
15: **if** $x \neq$ *sense* and *sense* $= \{A, B\}$ **then**
16: **if** $x = \{B\}$ **then**
17: **set** *bnum* $\mapsto$ *bnum* $\vee$ 0100 --*Node at boundary A*
18: **if** $x = \{A\}$ **then**
19: **set** *bnum* $\mapsto$ *bnum* $\vee$ 0010 --*Node at boundary B*
20: **if** $x = \varnothing$ **then**
21: **set** *bnum* $\mapsto$ *bnum* $\vee$ 1000 --*Node at boundary A and B*
22: **if** `ping` message received from all one-hop neighbors **then**
23: **if** *bnum* $\neq$ 0000 **then**
24: **become** BNDY
25: **else**
26: **become** DONE

BNDY
27: *Spontaneously*
28: **if** *bnum* contains >1 non-zero bit **then**
29: **set** *bnum* $\mapsto$ 1001 --*Boundary $A \cap B$ and not equal*
30: **send** (`rprt`, *bnum*) to *parent* --*Initiate message to sink*
31: **become** DONE

DONE
32: *Receiving* (`rprt`, b)
33: **if** $b \vee$ *bnum* $\neq$ *bnum* **then** --*Check for new data*
34: **set** *bnum* $\mapsto b \vee$ *bnum* --*Data aggregation*
35: **send** (`rprt`, *bnum*) to *parent* --*Forward aggregate data*

IDLE, BNDY
36: *Receiving* (`rprt`, b)
37: Defer processing event until node transitions to state DONE

of the intersection between A and B is expected to scale approximately in proportion to $O(\sqrt{|V|})^1$. Thus the overall communication complexity of the reporting stage is expected to scale approximately $O(\sqrt{|V|}\log|V|)$, depending on the specific scaling characteristics of the shapes.

Overall communication complexity provides a guide to the *global* resources required by the algorithm. However, a potentially more important measure of efficiency is *load balance*: the number of messages transmitted by individual nodes. Our algorithm is expected to achieve constant communication complexity, $O(1)$, with any node transmitting at most 5 messages: one `ping` message, and (due to data aggregation) up to a maximum of four `rprt` messages. The vast majority of nodes are expected to transmit only the one required `ping` message, and zero `rprt` message, as a result of the spatial filtering.

4. EXPERIMENTS

In this short paper, space restrictions do not allow a full discussion of the experimental evaluation. Instead, we briefly outline the key experimental approach adopted.

- The algorithm described in the previous section was implemented within the agent-based simulation system NetLogo.

- A series of randomized experiments was conducted using this system to test the efficiency of the algorithm in practice. The experiments used randomized networks and randomized spatial regions.

- Four different strategies were investigated, relating directly to the two underlying strategies used in the algorithm. Specifically, four versions of Algorithm 1 were generated: 1. Exactly as in Algorithm 1; 2. With data aggregation disabled; 3. With spatial filtering disabled; and 4. With data aggregation and spatial filtering disabled.

- Experiments evaluated the overall scalability of the algorithm, and showed that the both strategies of spatial filtering and data aggregation, individually and in concert, resulted in improved communication efficiency. A regression analysis of the messages observed for a range of network sizes confirmed that in practice the algorithm can operate close to the expected $O(\sqrt{n}\log n)$ scalability.

5. DISCUSSION AND CONCLUSIONS

This research has shown that the combined strategies of data aggregation and spatial filtering can allow the construction of an efficient decentralized algorithm for determining the topological relationship between two regions. Crucially, and in contrast to previous work in this area, the algorithm is able to operate without any quantitate spatial information, making it suitable for deployment in networks that have no access to coordinate location information. While the overall communication complexity is $O(|V|)$, the `ping` message might easily be included as part of the network initialization, expected to be required anyway for any network. Thus, the effective overhead of this algorithm is in the `rprt` messages, approximately $O(\sqrt{|V|}\log|V|)$.

Current work is investigating several ways in which this algorithm might be easily extended, for example to determining the topological relations between multiple, rather than just two regions. Another natural extension is to determine the topological relations between regions not homeomorphic to a disk (i.e., regions with holes and disconnected parts). Further, an important issue not addressed in this short paper is the fundamental relationship between topological relations for continuous regions and our finite granularity observations of those regions. This issue is also the subject of current research.

[1] This is true at least for non-fractal shapes. For fractal shapes, the number of nodes at the boundary will scale in proportion to $O(|V|^{0.5*D})$, where D is the fractal dimension of the shape, cf. [12].

Acknowledgements

This research was supported under Australian Research Council's Future Fellowship scheme (project numbers FT0990521, FT0990811, FT0991917).

6. REFERENCES

[1] S. Nittel, A. Stefanidis, I. Cruz, M. Egenhofer, D. Goldin, A. Howard, A. Labrinidis, S. Madden, A. Voisard, and M. Worboys, "Report from the First Workshop on Geo Sensor Networks," *ACM SIGMOD Record*, vol. 33, no. 1, pp. 141–144, 2004.

[2] D. Estrin, R. Govindan, and J. Heidemann, "Embedding the internet," *Communications of the ACM*, vol. 43, pp. 38–41, 2000.

[3] N. Lynch, *Distributed Algorithms*. San Mateo, CA: Morgan Kaufmann, 1996.

[4] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proc. 22nd International Conference on Distributed Computing Systems (ICDCS)*, pp. 575–578, IEEE, 2002.

[5] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A tiny aggregation service for ad-hoc sensor networks," in *Proc. 5th Symposium on Operating System Design and Implementation (OSDI)*, pp. 131–146, 2002.

[6] M. B. Greenwald and S. Khanna, "Power-conserving computation of order-statistics over sensor networks," in *Proc 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, (New York), pp. 275–285, ACM, 2004.

[7] R. Zheng and R. Barton, "Toward optimal data aggregation in random wireless sensor networks," in *Proc. 26th IEEE International Conference on Computer Communications (INFOCOM)*, (Washington, DC), pp. 249–257, IEEE, 2007.

[8] M. F. Worboys and M. Duckham, "Monitoring qualitative spatiotemporal change for geosensor networks," *International Journal of Geographical Information Science*, vol. 20, no. 10, pp. 1087–1108, 2006.

[9] N. Santoro, *Design and Analysis of Distributed Algorithms*. New Jersey: Wiley, 2007.

[10] M. Sadeq and M. Duckham, "Effect of neighborhood on in-network processing in sensor networks," in *GIScience 2008* (T. Cova, K. Beard, M. Goodchild, and A. Frank, eds.), vol. 5266 of *Lecture Notes in Computer Science*, (Berlin), pp. 133–150, Springer, 2008.

[11] M. Egenhofer and R. Fransoza, "Point-set topological spatial relations," *International Journal of Geographical Information Science*, vol. 5, no. 2, pp. 161–174, 1991.

[12] M. Duckham and J. Drummond, "Assessment of error in digital vector data using fractal geometry," *International Journal of Geographical Information Science*, vol. 14, no. 1, pp. 67–84, 2000.