

TRACKING AREAL OBJECT IDENTITY IN SNAPSHOT SEQUENCES

Mingzheng Shi and Stephan Winter

Department of Geomatics
The University of Melbourne
Victoria, 3010, Australia
m.shi@pgrad.unimelb.edu.au, winter@unimelb.edu.au

Geosensor networks are deployed to detect and track dynamic geographic phenomena, or objects, over space and time. Object identity is a unique characteristic to distinguish different spatial objects. Based on tracking and analysis of the relations of object identities for example different classes of events can be derived. This paper introduces a spatiotemporal data model for the storage and maintenance of areal object lifespan in a decentralized network. Different from previous work, our algorithm uses snapshot sequences to analyze the relations of spatial objects, and it allows both abrupt and incremental changes. In contrast to previous approaches the presented approach can deal with change beyond incremental change. Our model can also become an essential component for analyzing static topological relations of spatial objects in a decentralized manner, or for detecting dynamic topological changes.

1 INTRODUCTION

Our world is dynamic, and many geographic phenomena are continuously changing in time and space by different causalities. There are two ontologies that are frequently used for analyzing change, referred to SNAP and SPAN (Grenon and Smith, 2004). The SNAP ontology has a snapshot-based view, where entities are organized as temporal sequences of snapshots. Snapshots are commonly used in geographical applications. Conventional geographical information systems (GIS), for instance, acquire spatiotemporal data by remote or in-situ observations of typically relative fine spatial, but coarse temporal granularity, and thus, represent spatial phenomena as static snapshots (Peuquet, 2001). When comparing two snapshot sequences, changes may appear to be abrupt due to the low temporal frequency of observations. Moreover, interpolation of the snapshots is not always appropriate to reconstruct the underlying dynamic processes of reality. In this case, the snapshot approach is unable to trace events that occur between snapshots.

Event-based approaches in SPAN ontology then become promising in theoretical studies (e.g., Worboys, 2005). An event-based spatiotemporal data model, for example, is proposed by (Peuquet and Duan, 1995). The model only records the timestamp when change occur, therefore it can be understood as a chronicle-based model that is dual to the snapshot-based model (Galton, 2004). In this case, changes or events can be explicitly stored in a compacted timeline, and additionally the redundancy of spatiotemporal data in the snapshot-based model can be reduced. However, one of the issues that the event-based model needs to resolve is the spatial, temporal, and thematic granularities of observations. With certain spatial and temporal resolution of observation, events may not be recorded somewhere, sometime. Even if someone may argue that continuous observation is possible with future technologies, there is still the problem of redundancy of observations. Large amount of energy can be consumed to observe an environment where no change has occurred.

Therefore, there exist situations where data from applications, e.g., snapshots, cannot satisfy theoretic analysis, and theoretic

models, e.g., event-based model, are difficult to apply to real world applications. It is the emergence of new observation tools, e.g., wireless sensor networks (WSN), that bring the theory and application closer, since WSN can provide new options of spatial, temporal and thematic granularities of observations. A WSN is a network of computing devices that can collaborate via radio communications. A WSN is also a network of observation devices, since each node in the WSN is equipped with sensors that enable thematically fine-grained observation of the environments. A WSN is able to observe change in real-time, and hence, with almost any temporal granularity (but typically coarser spatial granularity).

In the context of geographic information science, snapshot-based approaches and event-based approaches have been used to model WSNs. Snapshot-based approach are commonly used in applications, (e.g., Wark et al., 2007), where sensor nodes are tasked to periodically sense and store snapshots of an environment by setting the sensor network to a certain temporal sensing resolution. Although a snapshot approach is more practical, theoretical studies, (e.g., Worboys and Duckham, 2006), are more interested in event-based model of WSNs due to its advantages of detecting salient changes or events. But, as discussed above, event-based approach is based on the underlying assumption of continuous observation, which is not practical in real world scenarios. Moreover, the redundancy of continuous observations becomes more critical in the case of WSNs, since WSNs are battery-powered and energy resources are highly constrained for sensor nodes.

In this circumstance, this paper develops a new spatiotemporal data model for WSNs that incorporates both snapshot-based and event-based approaches, such that the model can adapt to different spatial, temporal and thematic granularities of observations, can detect, store, and analyze changes or events, and can reduce the redundancy of spatiotemporal data. Since temporal granularity of observation is allowed in our model, geographic phenomena are represented as sequences of snapshots. Also, our spatiotemporal data model is an object-based model, i.e., the geographic phenomena are represented as objects or entities. We are primarily interested in areal objects or regions. Our model relies on areal object identities for spatiotemporal data representation. Thus, each areal object will be provided a unique identity, and our goal is to decentrally and dynamically trace the areal object identities in snapshot sequences over time.

This paper has three major contributions. Firstly, we introduce a new model that integrates both snapshot-based and event-based approaches for spatiotemporal data representation. Secondly, we develop an approach to decentrally store the completed lifespan of areal objects in a sensor network. Finally, a decentralized algorithm is designed to maintain the lifespan of areal objects in snapshot sequences.

2 RELATED WORK

Geographic phenomena may continuously change over time their location, size, orientation, or their character. The change investigated in this paper is the change of spatial entities. A change occurs whenever spatial entities possess different spatial attributes at different times (Galton, 2000). Grenon and Smith (2004) classify entities in the spatiotemporal world into two categories: one is continuants, and the other is occurrents. The study of this paper is on the perspective of continuants that exist at a given time at a given level of granularity and undergo different types of changes over time.

Hornsby and Egenhofer (1997) suggest a qualitative representation of change. Their notion of change is based on object identity, and a set of operations that either preserve or change identity. This paper applies their concept of identity in sensor networks for decentralized maintenance of areal object identities. Stell (2003) starts from a partially ordered time scale and dynamic objects with four fundamental operations on these objects (birth, death, merge and split), similar to Hornsby and Egenhofer (1997). He then introduces granularity of objects by two further operations, amalgamation and selection. This means his concept of granularity is based on sets. In contrast, the granularities discussed in this paper are related to observations. A discussion of spatiotemporal data model can be found in Abraham and Roddick (1999).

As discussed in Section 1, decentralized approaches in WSNs can be classified into snapshot-based approaches and event-based approaches. Lian et al. (2007) propose a gradient boundary detection approach, where snapshots of a monitored region are transmitted to a designated central computer via the *sink* of the network at certain time intervals. A snapshot at a given time t is abstracted as a contour map, which consists of gradient boundaries. Only the nodes on the gradient boundaries need to report to the sink, which constructs a contour map. A unique work for collecting snapshot data is proposed by Skraba et al. (2006). Their method is called a sweep, i.e., a wavefront that traverses the whole network and passes all nodes in the network exactly once. Sarkar et al. (2008) apply the sweep method to construct iso-contours in a continuous scalar field. Given the constructed iso-contours of a snapshot, users can acquire contours at given values or ranges. All the above approaches are limited to static snapshots. They cannot dynamically analyze snapshot sequences to derive salient changes.

Duckham et al. (2005) develops a model to track salient changes or events in the environment. They model a sensor network as a triangulation network. And the triangulation can change over time in response to the movement of spatial phenomena. Worboys and Duckham (2006) also use triangulations, and spatial objects is represented as a set of triangles. Changes of spatial objects are represented as the insertion or deletion of triangles from the set. However, their approach is limited to incremental changes, i.e., the change of a single triangle at each time step. In contrast, our approach allows both abrupt and incremental changes. Sadeq and Duckham (2008) investigate different sensor network structures for detecting topological events. The fundamental topological events include appearance, disappearance, merge and split, etc. Their approach also assume incremental changes. Their classification of events is based on the theoretical study of Jiang and Worboys (2009). Although there are a large amount of existing work on sensor network data models, this paper is the first work to investigate areal object lifespans in snapshot sequences.

3 SPATIOTEMPORAL DATA REPRESENTATION

For geographic data representation, there is a distinction between field-based and object-based models. Our spatiotemporal data model is an object-based model, and we use an object-oriented approach to implement our model. In our approach, a WSN is modeled as a set of point objects with point observations. An appropriate data structure is essential for efficient interaction or communication among point objects. The primary structure in our model is an edge. Edges are important components for spatial objects. For example, an area is embedded in a polygon that is constituted as a sequence of edges. By sensor network point observations, geographic phenomena will be abstracted as areal objects that consist of points, and these points will then be organized as sequence of edges.

Figure 1 illustrates an example about the representation of dynamic areal objects in a sensor network. Suppose Figure 1(a)-(d) are at four different time steps, e.g., t_1 , t_2 , t_3 , and t_4 . Between t_1 and t_2 , two polygons expand. Between t_2 and t_3 , the two polygons merge into a larger polygon. And the larger polygon expands again between t_3 and t_4 . If the temporal granularity of observation is fine enough to acquire all the four snapshots, then we can infer three changes or events, i.e., expand, merge, and expand, between the four snapshots respectively. If the given temporal granularity can only acquire two snapshots at t_1 and t_4 in Figure 1(a) and (d), then only one event, i.e., merge, can be inferred between the two snapshots. These changes between snapshots may include incremental changes, but there are also non-incremental changes, for example, the change between Figure 1 (c) and (d). By allowing granularity of observations in our model, we should also allow granularity of changes.

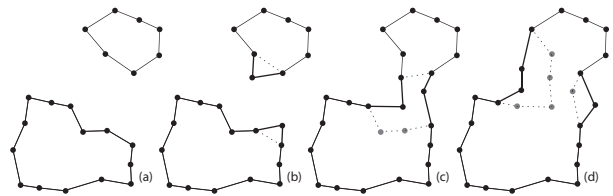


Figure 1: The evolving of spatial objects over time. (a)-(d) are at different time steps, e.g., t_1 , t_2 , t_3 , and t_4 .

There are always tradeoffs between granularity of observations and granularity of spatiotemporal data. If a finer granularity of observation is applied to a sensor network, more energy will be consumed for observation, communication, and data processing in the network, but the acquired data would have higher quality. On the other hand, coarser granularity of observation means less energy consumption and lower data quality. Our model can be applied to different granularity of observations. Note that there is limited number of point observations in a sensor network deployed area, which means spatial granularity of observation has a limit. Even the finest temporal and thematic granularity of observations are applied, inaccuracy and imprecision will be always associated with the observed spatiotemporal data. In this paper, we assume a certain granularity of observations can be decided for an application, so that the energy consumption and the accuracy of spatiotemporal data can be balanced.

This paper will focus on decentralized tracking of dynamic areal objects in a sensor network under certain temporal granularity of observations, as the example in Figure 1. Since temporal granularity of observation is taken into consideration, a sequence of snapshots will be acquired by sensor networks. Each snapshot in the sequence will contain a set of areal objects. By the evolution

of areal objects over time, such as expansion, contraction, split, and merge, edges of areal objects could remain unchanged, or be inserted or deleted. The insertion and deletion of edges are important for our data model. In the rest of this section, we will define the representation of inserted and deleted edges in a sensor network.

Firstly, a sensor network can be modeled as a directed planar graph $G = (V, E)$, which is built by Delaunay triangulation (Preparata and Shamos, 1985). In the graph G , V is a set of nodes and E is a set of edges, e.g., (v, v') , which represent the direct communication links between the nodes $v \in V$ and $v' \in V$. We assume that E is symmetric, i.e., if there are $v, v' \in V$, $(v, v') \in E$, then $(v', v) \in E$. Note that the direction of a directed edge will be illustrated in a figure when it is relevant, otherwise the representation of G can be simplified as in Figure 2(a).

The set of neighbors of $v \in V$ is denoted as $neighbor(v) = \{v' : (v, v') \in E\}$. For example in Figure 2(a), $neighbor(v) = \{a, b, c, d, e\}$, and the set $neighbor(v)$ is sorted into clockwise order. Our approach is a decentralized approach in the sense that each point object only stores information about itself and its immediate neighbors. Each point object has its own datasets, and global knowledge, e.g., the information about the whole network G , is unavailable in point objects.

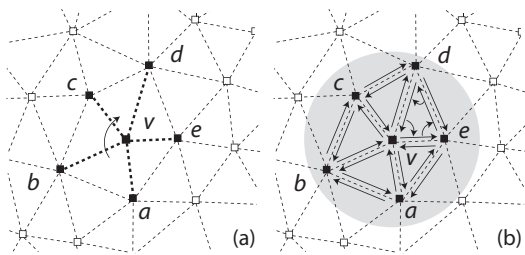


Figure 2: Sensor network structure. (a) An example of a communication graph G , where the set $neighbor(v)$ of each node v is sorted into clockwise order. (b) An areal object, illustrated as an solid circle, is represented by a sequence of boundary edges (a, b) , (b, c) , (c, d) , (d, e) , and (e, a) .

A node v and its neighbors are organized into *directed cycles*, or simply *cycles*. A cycle is denoted as (v, b, a, v) , where the three nodes v, a and b are distinct, and there is an edge for any two consecutive nodes in the cycle, i.e., (v, b) , (b, a) , $(a, v) \in E$. Examples of cycles can be found in Figure 2(b), where (v, b, a, v) and (v, c, b, v) are cycles. Since $neighbor(v)$ is in clockwise order, all the cycles are counterclockwise, and an example is shown in Figure 2(b). Based on the above definition, we can define the representation of areal objects in a sensor network:

Definition 1 A directed edge $(v, v') \in E$ is an object edge, if both v and v' are located in an areal object.

Definition 2 A cycle (v, b, a, v) is an object cycle, if all the vertices of the cycle, i.e., v, a, b , are located in an areal object.

Definition 3 An object edge $(v, v') \in E$ is a non-boundary edge, if the edge belongs to an object cycle.

Definition 4 An object edge $(v, v') \in E$ is a boundary edge, if it does not belong to any object cycles.

Areal objects are represented as sequence of boundary edges in the network. An example is shown in Figure 2(b), in which the nodes v, a, b, c, d and e are located in an areal object. The edges (a, b) , (b, c) , (c, d) , (d, e) , (e, a) are boundary edges, since they do not belong to any object cycles.

Since areal objects are evolving over time, sequences of boundary edges would dynamically change at different snapshots, as illustrated in Figure 3. The difference of boundary edges between two consecutive snapshots is represented by the insertion and deletion of boundary edges. If a boundary edge appears between t_{i-1} and t_i , then we will insert the boundary edge into relevant datasets with a timestamp of t_i , and vice versa. Each sensor node in the network may be involved in the insertion and deletion of boundary edges in its own datasets. To represent the change of areal objects in a sensor network, we require two definitions:

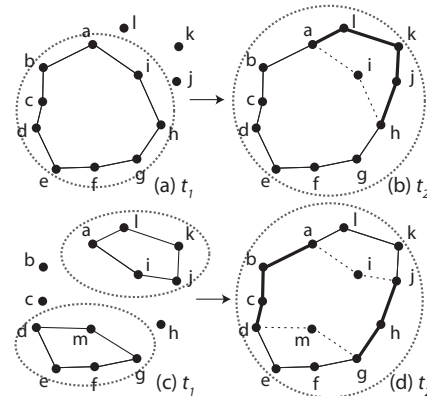


Figure 3: Examples of creation and deletion of edges. The created edges are denoted as thick continuous lines, and the deleted edges are denoted as dashed lines. Areal objects are marked by dashed-line ellipses. Note that the directions of edges are not shown in the figure.

Definition 5 A boundary edge $(v, v') \in E$ is an inserted edge, at time t_i if the edge is not a boundary edge at previous time t_{i-1} , but become a boundary edge at time t_i .

Definition 6 A boundary edge $(v, v') \in E$ is an deleted edge, at time t_i if the edge is a boundary edge at previous time t_{i-1} , but is not a boundary edge at time t_i .

The inserted and deleted edges will be decentrally stored at different sensor nodes in the network with a timestamp. Two local datasets, i.e., $insertedEdge(v, t)$ and $deletedEdge(v, t)$, are required at each node v to store inserted and deleted edges over time. For example, an area object is expanded from Figure 3(a) to (b), so that edges al, lk, kj , and jh have been inserted and edges ai and ih have been deleted. And the other edges remain unchanged. Six nodes a, i, h, j, k , and l need to record the insertion and deletion of boundary edges. For example, there are $insertedEdge(a, t_2) = \{(a, l)\}$ and $deletedEdge(a, t_2) = \{(a, i)\}$ for the node a .

4 AREAL OBJECT IDENTITY

4.1 Leader and Trajectory

In our model, spatiotemporal data is decentrally stored in each node in the network. These spatiotemporal data acquired at different time steps may need to be accessed for qualitative and

quantitative analysis. For example, two consecutive snapshots at time steps t_i and t_{i-1} could be used to calculate the area difference of an areal object between two time steps. This section introduces the concept of leader and trajectory, which will enable the decentralized retrieval of historical spatiotemporal data.

We define a leader of an areal object as a selected node that is currently located inside the areal object. A leader trajectory, or simply trajectory, is the path of the leader over time. A trajectory consists of a sequence of edges, called trajectory edges. Since areal objects are changing over time, the leaders and leader trajectories need to be dynamically updated. Figure 4(a)-(d) show a general idea about the update of leaders and leader trajectories. Assume that there is no areal object at initial time step t_0 . Between time step t_0 and t_1 , an areal object appears. A node a on the boundary edges of the areal object is selected as the leader. The leader a reports the appearance of the areal object to the sink by greedy routing (Karp and Kung, 2000). During the routing, the information of the leader a will be recorded along the path from a to the sink. We regard the path between the leader a and the sink as one part of the leader trajectory.

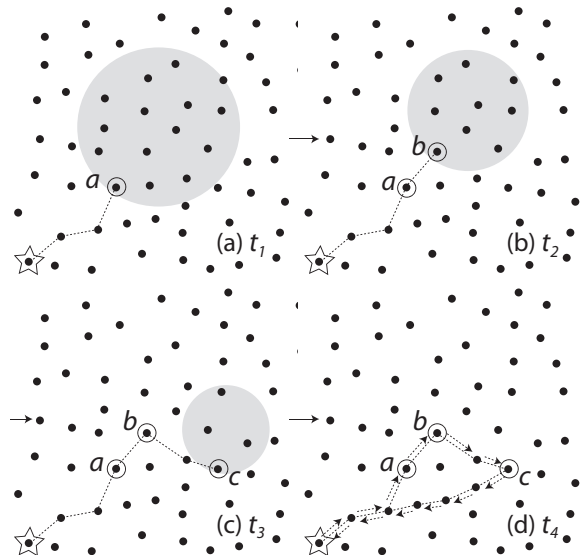


Figure 4: Leaders and leader trajectories need to be updated when areal object changes over time. Leaders are marked by circles, and dashed lines represent the leader trajectories.

As in Figure 4 (a) and (b), the areal object contracts between t_1 and t_2 , and the leader a is not longer in the areal object. The node b is then selected as the new leader, and the path between the two leaders a and b is added into the leader trajectory. Similarly, at t_3 in Figure 4(c), there is an update of the trajectory between leader b and c . Finally, between t_3 and t_4 , the areal object disappears. The leader c will report the disappearance of the areal object to the sink. And at the same time, the nodes along the path between c and the sink will record the trajectory. A completed trajectory that enables the access of the full history of the areal object is then formed at the sink. The completed trajectory is a closed traversable trail, as shown in Figure 4(d). The leader trajectory can be considered as a chronicle that links all the relevant snapshots of an areal object.

4.2 Trajectory Identity

To correctly access spatiotemporal data of areal objects, each leader trajectory should have an identity. Since each areal object has only one trajectory, the trajectory identity is also used as

the identity of the areal object. The identity of a trajectory will be decentrally stored in each trajectory edge. For example, in Figure 5(a), node a is a leader and a trajectory identity, e.g., A , is given to each edge on the trajectory. To correctly access each snapshot of an areal object from the trajectory, each boundary edge of the areal object at each snapshot should also have an identity. And the identity of boundary edges should match the identity of the trajectory. In Figure 5(a), each boundary edge of the areal object is also provided the same identity A .

The identity A will be used to identify the trajectory over time. Thus, when the areal object changes over time, the same identity should be maintained. In Figure 5(b), the areal object has contracted at t_2 , and the leader is changed from a to b . The same identity A should be updated for both new trajectory edges and new boundary edges, as shown in Figure 5(b).

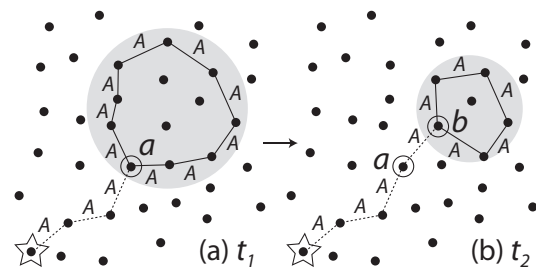


Figure 5: Trajectory Identity. Boundary edges of the areal object have the same identity as the trajectory edges.

As discussed in Section 3, each boundary edge of areal objects has a timestamp. For instance, the boundary edges in Figure 5 (a) and (b) are associated with a timestamp of t_1 and t_2 respectively. The snapshot sequences of areal objects can be easily retrieved using trajectory and its identities. For example, if a query inserted from the sink is interested in the areal objects at t_2 , then the relevant data can be retrieved by a traversal from the sink to the leader b . And the leader b (with identity A) can then traverse all the boundary edges that have a timestamp of t_2 and have the identity of A . Before the trajectories can be used for accessing spatiotemporal data in a decentralized network, we firstly need to maintain the trajectory identities when areal objects change over time.

4.3 Trajectory Maintenance

There are six types of changes of areal objects that require possible updates of trajectories. These six types of changes are: appearance, disappearance, expansion, contraction, merge, and split. When an areal object appears, a new trajectory needs to be created for the areal object, and the trajectory needs a unique identity. Figure 6 has a abstract representation of the changes of trajectories. In the figure, the star represents the sink, circles or nodes represent leaders, and a continuous line represents a path between two nodes. In Figure 6(a)-(b), an areal object appears, a leader is selected, and an identity of A is given to the leader. Also the path between the leader and the sink has the same identity A . An areal object with a trajectory identity A will be simply called areal object A .

As discussed in the previous section, when an areal object expands or contracts, the trajectory of the areal object may need to be updated. In Figure 6(c), a new leader is selected for the trajectory A due to the expansion of the areal object, and there is a path between the two leaders. Between t_2 and t_3 , areal object A contracts and a new areal object B appears, and their trajectories are updated correspondingly.

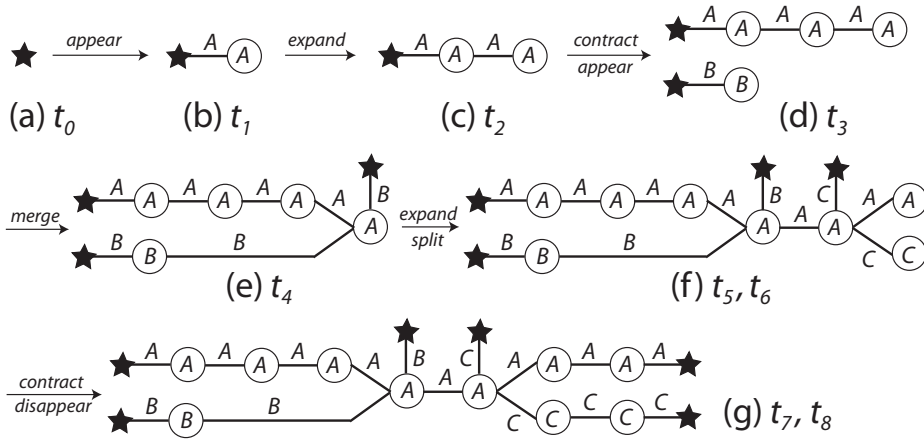


Figure 6: The trajectories need to be updated when areal objects change over time. The star represents the sink, different circles or nodes at one time step represent different leaders, and a continuous line represent a path between two nodes. There may be multiple stars at one time step, but these stars represent the same sink at a single location.

If two areal objects A and B merge, the two trajectories of the areal objects have to merge into one trajectory, as in Figure 6(e). One identity, e.g., A , can remain for the merged trajectory, and the other trajectory B is concluded to the sink. With the conclusion of trajectory B , the sink will also aware the merge of two areal objects A and B . In contrast, if an areal object A splits into two areal objects, the trajectory A has to split into two trajectories, e.g., A and C , as shown in Figure 6(f). In Figure 6(g), both areal objects A and C has disappeared, and both trajectories A and C conclude to the sink.

5 DECENTRALIZED ALGORITHMS

Since our algorithms are run decentrally in each sensor node in the network. The algorithm for updating trajectory identity will be dependant on different types of nodes in the network. We assume that each sensor node has only binary readings, i.e., 0 and 1. If a sensor node is located inside an areal object, its reading is 1, otherwise the reading would be 0. We define that a node is an *active* node if it changes readings between two consecutive time steps. We use the set $active(t)$ to represent all active nodes at the time step t .

In Figure 7 (a) and (b), for instance, an areal object expands from t_1 to t_2 , and there are inserted edges (c, d) , (d, e) , and (e, a) , deleted edges (c, v) and (v, a) , and active nodes d and e . We define another type of edges:

Definition 7 An inserted or deleted edge is a transition edge if the edge connects an active node and a non-active node.

For example, in Figure 7(b), the inserted edge (c, d) is a transition edge that has a initial node $c \notin active(t_2)$ and a terminal node $d \in active(t_2)$. Also (e, a) is a transition edge, but the edge begins at an active node e and ends at a non-active node a .

Transition edges are always in pairs. If there is a transition edge with an initial non-active node and a terminal active node, then there always exists another transition edge with an initial active node and a terminal non-active node. A pair of transition edges is always connected by a trial. For example, (c, d) and (e, a) are connected by the trial $c \rightarrow d \rightarrow e \rightarrow a$. The set of transition edges of a node v is denoted as $transitionEdge(v, t)$, for instance, $transitionEdge(c, t_2) = \{(c, d)\}$ in Figure 7(b).

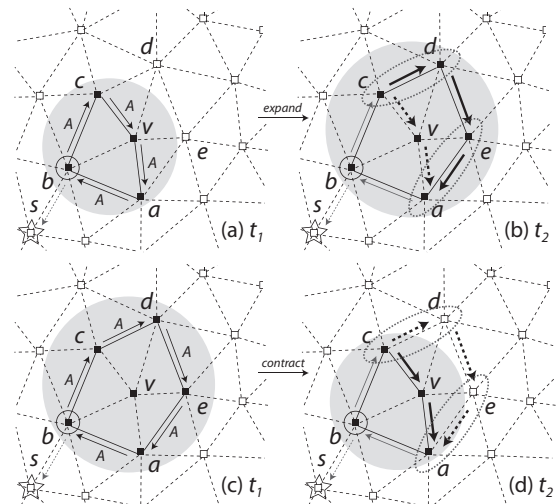


Figure 7: Expansion and contraction. Leaders are marked by circles, and transition edges are marked by dashed-line ellipses.

As discussed in Section 4.3, trajectories need to be updated differently when areal objects undergo different types of changes over time. Our decentralized algorithms should firstly distinguish different types of changes, so that trajectories can be update properly. A large number of nodes may be involved in the changes of areal objects, and thus in-network aggregations among relevant nodes are necessary. In our algorithm, graph traversals are used to organize the aggregation of decentralized spatiotemporal data. If all the relevant nodes initialize traversals in the network, there would be a large number of redundant traversals, and the efficiency of the algorithm will be significantly reduced. In this case, only a small subset of sensor nodes is eligible to initialize traversals.

In the algorithm, only the node v that has transition edges, i.e., $transitionEdge(v, t) \neq \emptyset$, will be eligible for starting traversals. And the traversals will always start at non-active nodes. For example, in Figure 7(b), $\{a, c\} \notin active(t_2)$ and $\{d, e\} \in active(t_2)$, so the traversal would start from non-active nodes a or c . Since node c is the initial node of the transition edge (c, d) , node c will start a traversal to visit all inserted edges, i.e., $c \rightarrow d \rightarrow e \rightarrow a$, and reach another non-active node a . Node a will then notice that c and a have the same identity A , and the

traversal has visited three inserted edges (c, d), (d, e), and (e, a). An expansion of the areal object A is detected at node a . Similarly, in Figure 7(d), the non-active node c will start a traversal to visit all deleted edges: $c \rightarrow d \rightarrow e \rightarrow a$. And node a will detect a contraction of areal object A .

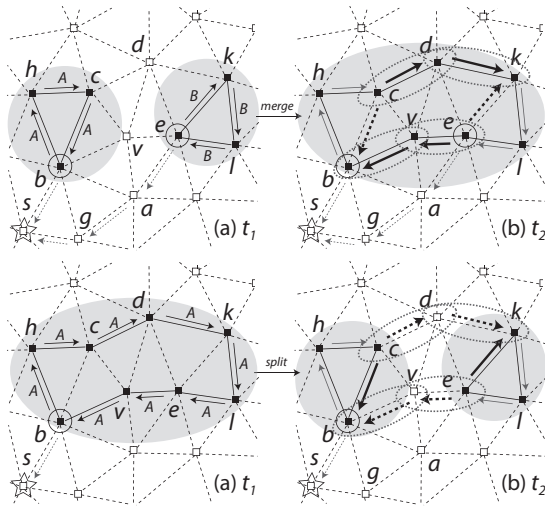


Figure 8: merge and split.

In Figure 8(b), two areal objects A and B have merged. There are two active nodes d and v and four transition edges (c, d), (d, k), (e, v), and (v, b). Both node c and e are eligible for starting traversals, since $\{c, e\} \notin \text{active}(t_2)$ and they are initial nodes of transition edges. The traversal starts from c will traverse inserted edges (c, d) and (d, k) and reach another non-active node k : $c \rightarrow d \rightarrow k$. The other traversal from e is $e \rightarrow v \rightarrow b$. Node k will notice c and k have different identities, and the traversal is detected at node k . Note that b can also detect the merge of two areal objects. In Figure 8(c) and (d), an areal object splits into two areal objects. Similarly, splits can also be detected by traversals.

6 CONCLUSION

This paper has introduced a new spatiotemporal data model that enables the storage of areal object lifespan in snapshot sequences. Each areal object has a unique trajectory that links all the relevant snapshots of the areal object together. The evolving of trajectories is dependent on the changes of areal objects. By the maintenance of trajectories in a decentralized network, different types of salient changes or events can also be derived by decentralized processing of snapshot sequences. An important area of future work is to use the areal object trajectories for qualitative and quantitative analysis of a static snapshot or dynamic snapshot sequences.

References

- Abraham, T. and Roddick, J., 1999. Survey of spatio-temporal databases. *Geoinformatica* 3(1), pp. 61–99.
- Duckham, M., Nittel, S. and Worboys, M., 2005. Monitoring dynamic spatial fields using responsive geosensor networks. In: *Proceedings of 13th Annual ACM International Workshop on Geographic Information Systems (GIS05)*, pp. 51–60.
- Galton, A., 2000. *Qualitative Spatial Change*. Oxford University Press.
- Galton, A., 2004. Fields and objects in space, time, and space-time. *Spatial Cognition and Computation* 4(1), pp. 39–67.
- Grenon, P. and Smith, B., 2004. Snap and span: Towards dynamic spatial ontology. *Spatial Cognition and Computation* 4(1), pp. 69–104.
- Hornsby, K. and Egenhofer, M., 1997. Qualitative representation of change. In: S. Hirtle and A. Frank (eds), *COSIT 1997, Volume 1329 of Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 15–33.
- Jiang, J. and Worboys, M., 2009. Event-based topology for dynamic planar areal objects. *International Journal of Geographical Information Systems* 23(1), pp. 33–60.
- Karp, B. and Kung, H., 2000. Gpsr: Greedy perimeter stateless routing for wireless networks. In: *MobiCom 00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ACM Press, New York, NY, USA, p. 243254.
- Lian, J., Chen, L., Naik, K., Liu, Y. and Agnew, G., 2007. Gradient boundary detection for time series snapshot construction in sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 18(10), pp. 1462–1475.
- Peuquet, D., 2001. Making space for time: Issues in space-time data representation. *Geoinformatica* 5(1), pp. 11–32.
- Peuquet, D. and Duan, N., 1995. An event-based spatiotemporal data model (estdm) for temporal analysis of geographical data. *International Journal of Geographical Information Systems* 9, pp. 7–24.
- Preparata, F. and Shamos, M., 1985. *Computational Geometry: An Introduction*. Springer, New York.
- Sadeq, M. and Duckham, M., 2008. Effect of neighborhood on in-network processing in sensor networks. In: T. C. et al. (ed.), *GIScience 2008, LNCS, volume 5266*, Springer, Heidelberg, pp. 133–150.
- Sarkar, R., Zhu, X., Gao, J., Guibas, L. and Mitchell, J., 2008. Iso-contour queries and gradient descent with guaranteed delivery in sensor networks. In: *Proceedings of the 27th Conference on Computer Communications*, IEEE, pp. 960–967.
- Skraba, P., Fang, Q., Nguyen, A. and Guibas, L., 2006. Sweeps over wireless sensor networks. In: *IPSN 06: Proceedings of the 5th international conference on Information processing in sensor networks*, p. 143151.
- Stell, J., 2003. Granularity in change over time. In: M. Duckham, M. Worboys and M. Goodchild (eds), *Foundations in Geographic Information Science*, Taylor & Francis, London, pp. 95–115.
- Wark, T., Corke, P., Sikka, P., Klingbeil, L., Guo, Y., Crossman, C., Valencia, P., Swain, D. and Bishop-Hurley, G., 2007. Transforming agriculture through pervasive wireless sensor networks. *IEEE Pervasive Computing* 6(2), pp. 50–57.
- Worboys, M., 2005. Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Systems* 19(1), pp. 128.
- Worboys, M. and Duckham, M., 2006. Monitoring qualitative spatiotemporal change for geosensor networks. *International Journal of Geographical Information Systems* 20(10), pp. 1087–1108.