Mike Worboys[*] and Matt Duckham[†]

[*]National Center for Geographic Information and Analysis
University of Maine, Orono, ME 04469, USA
Tel: +1 207 581 3679
Fax +1 207 581 2206
worboys@spatial.maine.edu

[†]Department of Geomatics
University of Melbourne, Victoria, 3010, Australia
Tel: +61 3 8344 6935
Fax +61 3 9347 2916
mduckham@unimelb.edu.au

Monitoring qualitative spatiotemporal change for geosensor networks

Abstract. Recent technological advances in geosensor networks demand new models of distributed computation with dynamic spatial information. This paper presents a computational model of spatial change in dynamic regions (such as may be derived from discretizations of continuous fields) founded on embeddings of graphs in orientable surfaces. Continuous change, connectedness, and regularity of dynamic regions are defined and local transition rules are used to constrain region evolution and enable more efficient inference of a region's state. The model provides a framework for the detection of global high-level events based on local low-level "snapshot" spatiotemporal data. The approach has particular relevance to environmental monitoring with geosensor networks, where technological constraints make the detection of global behavior from local conditions highly advantageous.

Monitoring qualitative spatiotemporal change for geosensor networks

Abstract. Recent technological advances in geosensor networks demand new models of distributed computation with dynamic spatial information. This paper presents a computational model of spatial change in dynamic regions (such as may be derived from discretizations of continuous fields) founded on embeddings of graphs in orientable surfaces. Continuous change, connectedness, and regularity of dynamic regions are defined and local transition rules are used to constrain region evolution and enable more efficient inference of a region's state. The model provides a framework for the detection of global high-level events based on local low-level "snapshot" spatiotemporal data. The approach has particular relevance to environmental monitoring with geosensor networks, where technological constraints make the detection of global behavior from local conditions highly advantageous.

1   Introduction and background

This paper presents a computational model of spatial change that can be used to represent and distinguish certain kinds of spatial events. The spatial objects involved in these events are regions, for example derived from discretizations of continuous fields. The overall aim of this research is to develop a framework that can bridge the gap between the low-level detail of spatiotemporal information systems and users' high-level domain conceptualizations in terms of change, process, and event. At the same time, the framework is explicitly computational, being discrete and compatible with efficient, local processing.

The goal of efficient, local detection of change is motivated primarily by technological developments in *geosensor networks*. A geosensor network is a distributed ad-hoc wireless network of sensor-enabled miniature computing platforms (a *sensor network*) that monitors phenomena in geographic space (Nittel et al., 2004). The model developed in this paper is designed to be applicable to geosensor networks, where the generation of high-level qualitative knowledge about complex dynamic geographic phenomena, based on real-time sensor data, is essential to spatial decision support. However, the model is also intended to be general enough to be efficiently applied across a range of computational environments, including cellular automata and conventional spatiotemporal databases.

This work provides a computational model of spatiotemporal data that enables us to represent and query events on regions. Our approach is first to formally define a simple, discrete model of space (section 3) and show how this model can be used to represent continuous change, regularity, and connectedness in dynamic regions (section 4). Using local transition rules to constrain change, several different representations of continuous change can be realized

(section 5). In turn, this enables different types of spatial events to be detected (section 6) for one or more regions.

In summary, the primary contributions of this paper are the development of a model of space-time that:

- enables the representation and definition of different types of spatial events resulting from continuous changes in the real world;
- is computationally oriented, being discrete and compatible with efficient and local processing; and
- helps bridge the gap between of high-level conceptualizations of events and low-level "snapshot" spatiotemporal data.

1.1 Motivation

As an example, consider the changes that might occur during the lifetime of an oil slick, following an accidental oil tanker collision. The oil slick will *appear* at the point in time when the accident occurs. It is likely that the slick will then rapidly *expand*. Ocean currents may cause a *hole* to form in the slick, later on leading slick to break up and *split* into different parts. As the oil slick is contained and treated, the different parts may begin to *contract* and ultimately *disappear*. These events are the results of continuous changes in the oil slick, but data about the extent of an oil slick can only be captured in the form of discrete snapshots at particular points in time and space (i.e., from sensor readings). Further, the key events italicized above are not explicitly represented by the spatiotemporal snapshot data. A pollution management team might deploy a dense array of sensors throughout the vicinity of slick in order to provide real-time information about when and where these different events are occurring (e.g., when the slick started to contract and where the slick split).

Although this motivational scenario is somewhat beyond the capabilities of today's technology, the development of similar applications of geosensor networks are a topic for current research (e.g., Xu and Lee, 2005). The model presented in this paper provides a basis for efficiently deriving salient events using local processing of spatiotemporal information within the geosensor network itself.

2 Background

The background to this work is the evolution of formal and computational models of dynamic spatial phenomena (2.1) and the state-of-the-art in geosensor network technology (2.2).

2.1 Dynamic spatial phenomena

Current spatiotemporal information systems typically represent change implicitly using a series of static *snapshots*. The general forms of queries to such systems are "What was the state of this object at that time?" or "At what time did

this object have that state?" (Abraham and Roddick, 1999; Snodgrass, 1995; Worboys, 1994). An objective of recent research is the explicit representation of spatial *events* (Hornsby and Egenhofer, 2000; Worboys, 2005; Worboys and Hornsby, 2004). Galton (2004) makes the distinction between *histories*, which are functions from a temporal domain to attribute values, or properties of objects, and *chronicles*, which treat dynamic phenomena as collections of happenings. From an ontological perspective, we can make an initial division of entities that exist in the world into *continuants* that endure through time (e.g., oceans, oil tankers, and oil slicks), and *occurrents* that happen or occur and are then gone (e.g., ocean tides, currents, or the splitting of an oil slick into disconnected regions). Grenon and Smith (2004) call temporal sequences of object configurations the SNAP ontology, and the event/action view, the SPAN ontology. It is the identification of SPAN entities in the evolution of spatial regions that is the focus of this paper.

2.2   Geosensor networks

As defined above, a geosensor network is a distributed ad-hoc wireless network of sensor-enabled miniature computing platforms that monitors phenomena in geographic space (Nittel et al., 2004). Individual sensor nodes are low cost and low power potentially allowing dense networks of nodes to be deployed to monitor environmental phenomena. Today's geosensor networks are relatively small, comprising tens or hundreds of sensors. For example, networks of between 50 and 150 nodes have been used to monitor habitat micro-climates for a rare species of petrel in Maine's Great Duck Island (Szewczyk et al., 2004a,b). Such geosensor networks provide the capability to monitor geographic phenomena in remote, sensitive, or hazardous environments at much higher spatial and temporal granularity than possible with conventional monitoring technology. In the near future, geosensor networks will comprise thousands or millions of geosensors (e.g., "smart dust", Kahn et al., 1999).

There are many research and technological challenges to overcome before reliable geosensor networks of two or three orders of magnitude larger than today's networks become a reality. Current research into geosensor networks is proceeding rapidly on several fronts. For example, power consumption is an overriding factor in many aspects of geosensor network design. The limited battery power of untethered sensor nodes means energy conservation is especially important with respect to power-intensive operations like communication (Singh et al., 1998). Another area toward which a great deal of activity has been directed is the design of efficient and robust routing algorithms for geosensor networks (e.g., Karp and Kung, 2000; Perkins et al., 2001; Braginsky and Estrin, 2002). Finally, determining the spatial location of nodes (termed *localization*) also presents particular problems to geosensor networks (Akyildiz et al., 2002). Conventional geodetic positioning systems, like GPS, require too much power and are not robust enough for geosensor networks (e.g., GPS doesn't operate indoors or outdoors where the GPS signal is occluded by a "radio shadow").

Underlying much research into geosensor networks is the aim of "decentralized coordination with local decision making to achieve the intended global goal" (Estrin et al., 2000). Decentralization has at least two major benefits. First, decentralized coordination implies lower communication overheads, and thus lower power consumption. Second, decentralization simplifies geosensor network algorithms and architectures, making them more scalable, reliable, and robust. The overall approach proposed in this paper is in keeping with the aim of greater decentralization in geosensor networks. However, unlike most research in the domain of geosensor networks, our model is explicitly spatial and temporal, enabling the identification of salient qualitative spatial events in dynamic spatial phenomena.

3   Combinatorial maps and triangulations

This section develops the formal bases for evolving spatial structures in two dimensions. The foundations of this work are embeddings of graphs in orientable surfaces. An *embedding* of a graph in a surface is an assignment of its nodes and edges to points and paths on the surface. The assignment is a graph isomorphism, and paths must only intersect at embedded nodes. In the case that the surface is orientable, an orientation can be assigned to any of its points. This orientation can be used to provide an (anti-clockwise) cyclic order of embedded edges around each embedded vertex. The additional information that the cyclic orders provides enable us, roughly, to characterize the embedded structure up to homeomorphism. In the work that follows, all the embeddings are triangulations of the surface.

The discrete structure we use to formally model this embedding is the *combinatorial map*. The combinatorial map was first introduced by Edmonds (1960) and developed by Tutte (1984), Guibas and Stolfi (1985), and others (e.g., Dufourd and Puitg, 2000).

Definition 1. *A (2-dimensional) oriented combinatorial map, or just* map, $\mathcal{M}$, *is a triple* $\langle D, \Theta_0, \Theta_1 \rangle$, *where D is a finite set of elements, called* darts, *Θ_0 is an involutory bijection on D (i.e., $\Theta_0^2 = 1$), and Θ_1 is a bijection on D. We may also assume that Θ_0 has no fixed points.*

In terms of a geosensor network, each dart d can be thought of as the capability of a sensor for a potential communication with partner within its communication range. The dart $\Theta_0 d$ represents the complementary capability of its partner. The function Θ_1 gives rise to a permutation that can be used to represent the sensor nodes themselves. Each permutation contains a counter-clockwise, cyclic ordering of darts around a node.

Figure 1 shows a simple example of a combinatorial map where $D = \{1, ..., 14\}$. In this example, Θ_0 and Θ_1 are given in cyclic notation as:

$$\Theta_0 = (1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12)(13\ 14)$$
$$\Theta_1\ = (1\ 10\ 11)(2\ 3)(4\ 12\ 13\ 5)(6\ 7)(9\ 8\ 14)$$
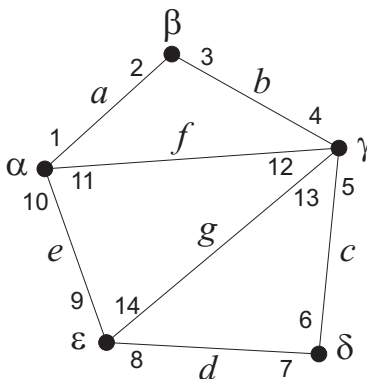
Fig. 1. Combinatorial map example

It can be seen that Θ_0 partitions the set of darts into sets of pairs of darts, and each such pair is called an *edge* of map $\mathcal{M}$. In our example, the edges are $a = \{1, 2\}$, $b = \{3, 4\}$, $c = \{5, 6\}$, $d = \{7, 8\}$, $e = \{9, 10\}$, $f = \{11, 12\}$, and $g = \{13, 14\}$. It is also clear that each of the cycles of Θ_1 represents a *vertex* of $\mathcal{M}$. In the example, the vertices are $\alpha = (1\ 10\ 11)$, $\beta = (2\ 3)$, $\gamma = (4\ 12\ 13\ 5)$, $\delta = (6\ 7)$, and $\epsilon = (9\ 8\ 14)$. We note that each dart uniquely determines an edge and a vertex.

It is straightforward to use Θ_0 and Θ_1 to calculate the ordering of edges round faces in a combinatorial map. The cycles of the composition $\Theta_0\Theta_1$ gives the ordering of darts, and converting the darts to their (unique) associated edges gives the ordering of edges. In the example above:

$$\Theta_1\Theta_0 = (1\ 3\ 12)(11\ 13\ 9)(14\ 5\ 7)(10\ 8\ 6\ 4\ 2)$$

and so the faces are bounded by edges *abf*, *fge*, *gcd*, and *gfedcba*. In general, a *face* of map $\mathcal{M}$ is a cycle of edges associated with a cycle of darts in $\Theta_1\Theta_0$. Alternatively, focusing on vertices rather than edges, we can consider the cycle of vertices (uniquely) associated with the edges to be the face. In our example, the faces are $\alpha\beta\gamma$, $\alpha\gamma\epsilon$, and $\gamma\delta\epsilon$.

Let $E = E(\mathcal{M})$ be the set of edges and $F = F(\mathcal{M})$ be the set of faces of $\mathcal{M}$.

Definition 2. *Let $\mathcal{M}$ be a given combinatorial map. A* triangle *in $\mathcal{M}$ is a face in $F = F(\mathcal{M})$ that is a 3-cycle of edges associated with a 3-cycle of darts in $\Theta_1\Theta_0$. Alternatively and equivalently, a triangle is a 3-cycle of vertices associated with a 3-cycle of darts in $\Theta_1\Theta_0$.*

Definition 3. *A* triangulation *Δ is a combinatorial map $\mathcal{E}$ which has the property that every edge in E belongs to either one or two triangles.*

From now on we will work with triangulations rather than more general combinatorial maps. Suppose from now on that our underlying triangulation is Δ.

Definition 4. *A (triangulated) region R in Δ is a subset of triangles of Δ. The set of all regions in Δ is denoted R(Δ).*

Definition 5. *Let ε be an edge in Δ and R be a triangulated region in Δ. ε is an* interior edge *of R if it belongs to two distinct triangles in R. ε is a* boundary edge *of R if it belongs to exactly one triangle in R. Otherwise, ε is an* exterior edge *of R.*

Definition 6. *Let β be a vertex in Δ and R be a triangulated region in Δ. β is an* interior vertex *of R if it belongs to a triangle in R and is incident with no boundary edges of R. β is a* boundary vertex *of R if it is incident with at least one boundary edge. Otherwise, β is an* exterior vertex *of R.*

Definition 7. *Let δ be a triangle in Δ and R be a triangulated region in Δ. δ is an* interior triangle *of R if δ ∈ R and all edges of δ are interior edges of R. δ is a* boundary triangle *of R if δ ∈ R and at least one of δ is a boundary edge of R. Otherwise, δ is a* exterior triangle *of R.*

Definition 8. *A region R is* regular *if each boundary vertex of R is incident with exactly two boundary edges of R.*

Note regularity rules out cases where triangles meet only at their vertices.

4 Dynamic regions arising from spatiotemporal fields

Up to now, the constructions apply to triangulations on any orientable surface. We will now be specifically concerned with triangulations embedded in the Euclidean plane. We also introduce into the formal model the notion that the triangulations can evolve through time. This will happen if we are modeling dynamic regions that arise from changing geographic phenomena, represented by spatiotemporal scalar fields (for example, spatiotemporal variations in levels of carbon dioxide in the atmosphere).

Formally, an *ST scalar field* is a function

$$f : S \times T \to D$$

where S, T, and D, are spatial, temporal, and scalar value domains, respectively. With our focus on spatial variation with time, we can restructure this field as

$$f : T \to (S \to D)$$

so, for each moment in time, we have a spatial variation of values from the value domain.

Assume that S is discretized as a triangulation framework Δ, as defined above. We want to view the state of the discretized field at any moment in time as an assignment of values from a discretized version of D to each of the triangles. In the simplest case, there is an assignment of values from $\{0, 1\}$ to each of the triangles in the triangulation Δ imposed by the field (this may be implemented through a threshold value, for example). This function can be used to induce a region R (a subset of triangles of Δ). Note that we have slightly overloaded our notation here, since R is now a function from T to a region of Δ.

4.1 Continuity and incremental change

We want a region R to evolve continuously through time in order to identify salient changes to R. An initial thought is to make the field function f continuous in the usual sense. However, this does not necessarily lead to continuously evolving regions. As a trivial counterexample, consider a field that is constant throughout the space and whose scalar values are continuously increasing with time. At some moment, all values may rise to our threshold value for detection of the field. Thus, the corresponding discretized region goes from empty to the universe in one jump. In order to avoid this, our definition of continuity works on the evolving region itself. The following definitions will also work with stand-alone triangulated regions, and it is not a requirement of this model that a region be derived from an underlying field.

Let $R : T \rightarrow R(\Delta)$ be a temporally evolving region. Suppose R only undergoes a finite number of changes during its evolution. The evolution can be modeled by the sequence $R_1, ..., R_n$.

Definition 9. $R = [R_1, ..., R_n]$ *is* continuous *if*

1. *For all* $i \in \{1, ..., n-1\}$ *no interior edge of* R_i *is an exterior edge of* R_{i+1}
2. *For all* $i \in \{1, ..., n-1\}$ *no exterior edge of* R_i *is an interior edge of* R_{i+1}

Definition 10. $R = [R_1, ..., R_n]$ *is* incremental *if*

1. *For all* $i \in \{1, ..., n-1\}$ R_i *and* R_{i+1} *differ by the insertion or deletion of a single triangle.*

Note that incremental change is a special case of continuity. Figures 2 and 3 show examples of continuous but not incremental change (figure 2), and incremental change (figure 3) to a triangulated region.

R_i $\qquad\qquad$ R_{i+1} $\qquad\qquad$ R_{i+2} $\qquad\qquad$ R_{i+3}
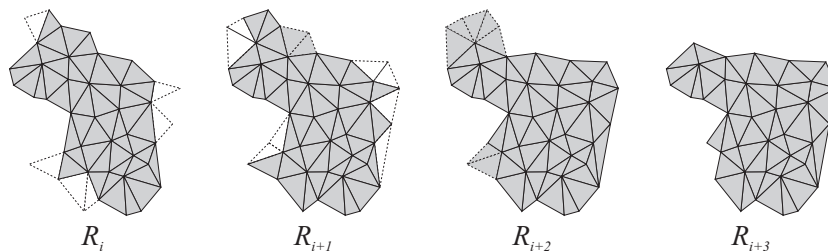
Fig. 2. Continuous but not incremental change (triangles to be added or removed indicated with dashed boundary)
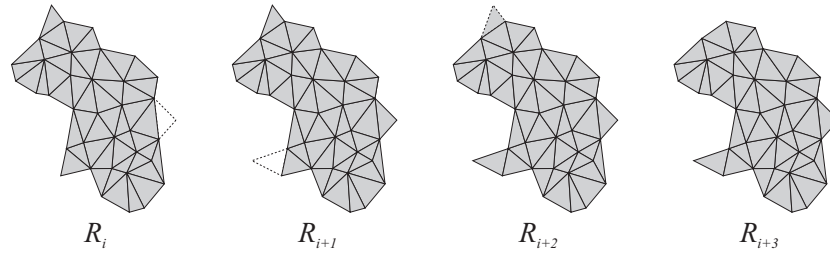
Fig. 3. Incremental change (triangle to be added or removed indicated with dashed boundary)

4.2 Connectedness and simple connectedness

A region $R \in R(\Delta)$ is *path connected* (or just *connected*) if for any two vertices v, v' of R, there exists a path from v to v' entirely contained within R. We can distinguish two kinds of path connectedness of relevance to this discussion, *weak path connectedness* and *strong path connectedness* (Worboys and Bofakos, 1993) (again, we omit the word "path" if no ambiguity results). In strong connectedness, there is a path from v to v' such that every point on the path, with the possible exception of the beginning and end points, can be surrounded by an open set (in the usual topology of the embedding plane) entirely contained within R (figure 4a); with weak connectedness, this condition is not satisfied (figure 4b). A triangulation R is *simply connected* if any simple closed path in R can be shrunk to a point continuously in R. As before, continuity is with respect to the usual topology of the plane. Intuitively, a region on a surface is simply connected if it contains no holes. A simply connected triangulation does not need to be regular. We may note that any regular, connected triangulation is also strongly connected. However, an irregular connected triangulation may also be strongly connected (see figure 4a).
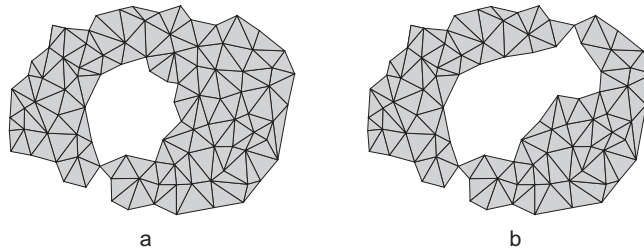

Fig. 4. Strongly-connected (a) and weakly connected (b) irregular triangulated regions

Computationally, determining whether a region R is connected can be achieved by considering the connectivity of the underlying graphs for R. For

example, if a standard graph search (for example, a breadth-first search) of the nodes and edges of the region R reveals that the graph is connected, then it follows that the region R is also connected. Using the dual adjacency graph formed by the set of faces (nodes) and adjacency between faces (edges) for R we can also check for weak connectivity. If R is connected, but the dual adjacency graph is not connected then R must be weakly connected. Conversely, if R is connected, and dual adjacency graph is connected then R must be strongly or simply connected. Distinguishing between strong and simple connection may require a more sophisticated test, but as we shall see in later sections, such properties may be inferred under certain conditions.

5 Region evolution

In this section we start to consider the sorts of changes that can occur to an evolving region observed at two proximal moments. In all cases, only a single triangle at a time is allowed to be inserted or deleted: the region is assumed to evolve incrementally (and therefore continuously). Future work will also address modeling continuous but not incremental evolution of regions. Although the region changes only incrementally, we assume that the temporal granularity of updates to the region is fine enough such that the region closely approximates changes in the underlying field itself. The aim of "tracking" the development of field with discrete elements of the triangulation is illustrated in Figure 5, where the region of above threshold activity in a continuously evolving field is captured by discrete incremental changes to the triangulated region, shown at three stages, R_5, R_{64}, and R_{115}.

R_5 R_{64} R_{115}
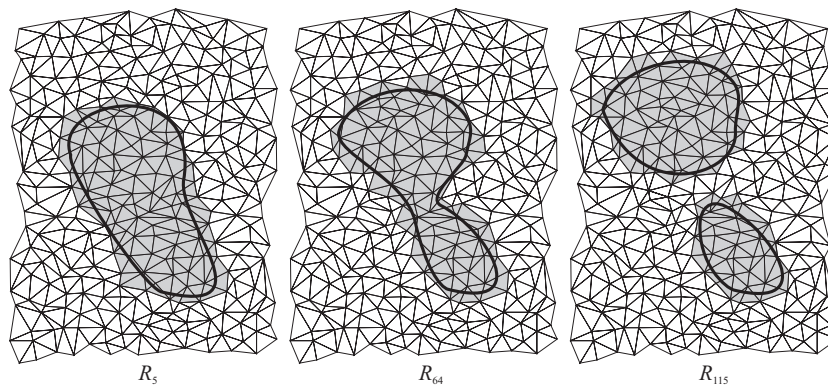
Fig. 5. Tracking the continuous evolution of a field

Incremental change constrains the evolutions of a dynamic region. In order to enable efficient local monitoring of salient changes, we may wish to further constrain the allowable changes to a region. A computationally efficient way of

doing this is to define local conditions that must be met in order for a specified change to occur. We refer to these local rules as *transition rules*, as they are akin to the transition rules found in cellular automata.

In the context of geosensor networks, the overall objective is to embed transition rules locally within each geosensor. In essence, these transition rules provide each triangle with global constraints to change. In turn, these global constraints can enable local inferences about the evolution of a region being monitored. The following section provides examples of four different transition rules.

5.1 Example transition rules

Four possible transition rules are considered in this section. Other transition rules might also be devised; the four transition rules in this section are chosen simply to illustrate the utility of the approach. These transition rules are combinations of either constraining or not constraining insertions and deletions to occur only at the boundary, and subject to preserving or not preserving regularity (see table 1). All four transition rules assume incremental change and so lead to continuously evolving triangulation in the sense of definition 9.

	Boundary change	General change
Regularity preserving	Transition rule A	Transition rule B
General	Transition rule C	Transition rule D

**Table 1.** Classification of the four transition rules

We may summarize the general form of the transition rules in terms of a simple finite state automaton for an individual triangle $\delta \in \Delta$ (figure 6). A triangle δ is either in a region R (state q_1) or not (state q_0). The transition rules define the conditions under which a triangle may be added or deleted from the region. The state of the region is defined by summing over the combined state of all the triangles $\delta \in \Delta$ that are in the region at any point in time.

Unlike a conventional cellular automata, one of the conditions that governs insertion or deletion of triangles from a region changes is the underlying state of the field being monitored (i.e., whether or not the environmental phenomenon has been detected within the triangle).

Transition rule A: Boundary change to regular R preserving regularity

Insertion conditions: $\delta \in \Delta \backslash R$ will be inserted into R to produce R' if:

1. The monitored phenomenon is detected in δ;
2. δ is edge-adjacent to one or two existing boundary triangles of R; and
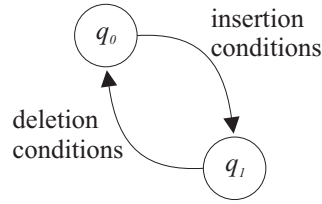
Fig. 6. Transition rules define changes of state between triangle $\delta \in \Delta$ being in (q_1) or out (q_0) of region $R \in R(\Delta)$

3. each vertex of δ that is a boundary vertex of R' is incident with exactly two boundary edges in R'.

Deletion conditions: $\delta \in R$ will be deleted from R to produce R' if:

1. The monitored phenomenon is not detected in δ;
2. δ is a boundary triangle of R; and
3. each vertex of δ that is also a vertex of R' is incident with exactly two boundary edges in R'.

Transition rule B: Change to regular R preserving regularity

Insertion conditions: $\delta \in \Delta \backslash R$ will be inserted into R to produce R' if:

1. The monitored phenomenon is detected in δ; and
2. each vertex of δ is incident with exactly zero or two boundary edges of R'.

Deletion conditions: $\delta \in R$ will be deleted from R to produce R' if:

1. The monitored phenomenon is not detected in δ; and
2. each vertex of δ that is also a vertex of R' is incident with exactly two boundary edges in R'.

Transition rule C: Unconstrained boundary change to R

Insertion conditions: $\delta \in \Delta \backslash R$ will be inserted into R if:

1. The monitored phenomenon is detected in δ; and
2. δ is edge-adjacent to at least one existing boundary triangle of R.

Deletion conditions: $\delta \in R$ will be deleted from R if:

1. The monitored phenomenon is not detected in δ; and
2. δ is a boundary triangle of R.

### Transition rule D: Unconstrained change to R

Insertion condition: $\delta \in \Delta \backslash R$ will be inserted into R if:

1. The monitored phenomenon is detected in δ.

Deletion condition: $\delta \in R$ will be deleted from R if:

1. The monitored phenomenon is not detected in δ.

Figure 7 graphically summarizes the transition rules, by depicting some of the allowable changes to an example regular simply connected region for transition rules A–D.
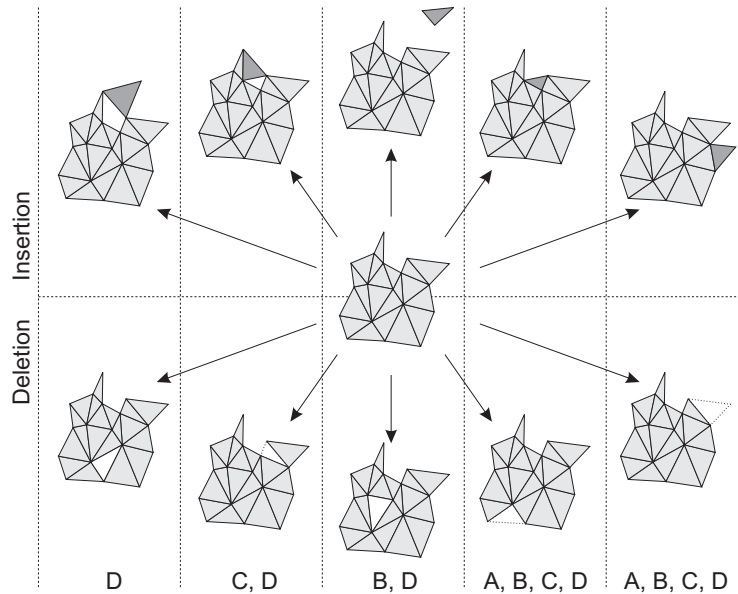
Fig. 7. Example allowable changes to simply connected region for transition rules A–D

Note that the transition rules are purely local to δ, the putative triangle for insertion or deletion. Thus, the transition rules potentially can provide a computationally efficient mechanism for keeping track of the connectedness of a region. For example, suppose that the state of a region $R(t)$ is known to be regular simply connected. For a time t'' where $t < t''$, if we know that for all t' such that $t < t' < t''$, 1) $R(t')$ is always regular, and 2) $R(t')$ has evolved through the application of only transition rules A and/or C, then it follows that $R(t'')$ is connected. Since maintenance of regularity can be checked using local conditions (see definition 8) and the transition rules themselves are purely local, such inferences may be an efficient way of determining whether a region is connected without the need to recompute connectivity.

Clearly, such inferences have direct application to efficient querying of geosensor networks, and related work is ongoing of the algorithms for detecting change

in the context of geosensor networks (see section 7.1). Recall that connectivity is a property that is not locally computable (section 4.2). Potentially connectivity could be computed using standard graph-based algorithms, like a depth first search of the dual graph of the triangulation. Such algorithms are not especially computationally expensive (e.g., optimal time complexity for depth first search is $O(|V| + |E|)$ where $|V|$ is the number of vertices and $|E|$ the number of edges in the in the triangulation). However, depth first search relies on exploring the entire triangulation, a very costly operation in the context of geosensor networks where communication is one of the most battery-draining operations for a sensor.

5.2 Conceptual neighborhoods

For each of the four transition rules A–D, we can summarize the possible changes in a non-empty dynamic region using a *conceptual neighborhood* diagram (see Freksa, 1991). In such a structure, two states are conceptual neighbors if it is possible to pass from one to the other, by whatever transformations are under consideration, *without passing through any intervening states.* The nodes of each conceptual neighborhood diagram represent jointly exhaustive and pairwise disjoint possible states for a dynamic region, based on its regularity and connectedness properties. The connections in the conceptual neighborhood diagram represent the possible transitions for a dynamic *region.*
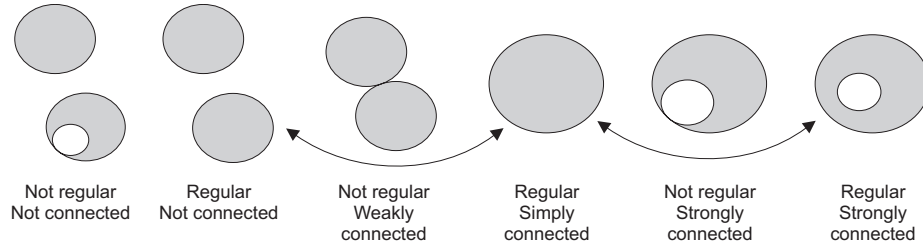
The conceptual neighborhood diagrams resulting from the different transition rules contain six possible states: not regular and not connected; regular and not connected; not regular and weakly connected; regular and simply (and strongly) connected; not regular and strongly connected; and regular and strongly (but not simply) connected regions. In the underlying field the distinction between some of these states, such as weak connection and strong connection, may not be especially salient. However, as we shall see, distinguishing between these states in the discrete region is useful as it enables different region transitions to be discerned apart.

The conceptual neighborhood diagram for transition rule A is shown in figure 8. Region transitions between the six states using transition rule A are not possible, so the conceptual neighborhood diagram for transition rule A is somewhat trivial and is included for completeness only.

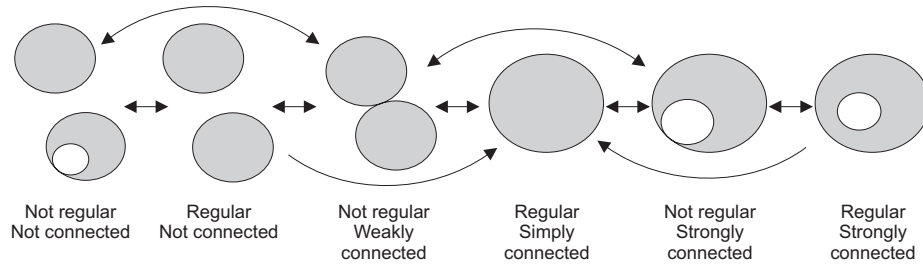| Not regular Not connected | Regular Not connected | Not regular Weakly connected | Regular Simply connected | Not regular Strongly connected | Regular Strongly connected |
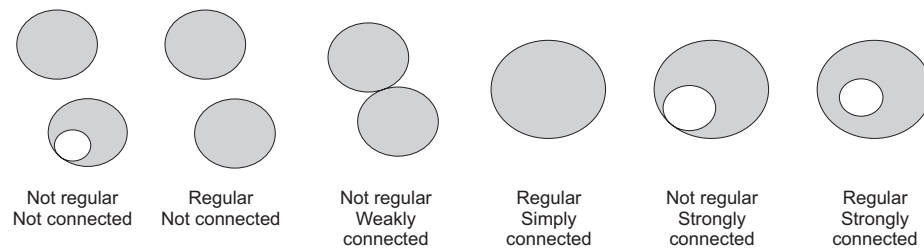
Fig. 8. Conceptual neighborhood diagram for transition rule A

The conceptual neighborhood diagram for transition rule B contains two region transitions: between regular, simply connected regions and regular, disconnected regions; and between regular, simply connected regions and regular, strongly connected regions (figure 9).

| Not regular | Regular | Not regular | Regular | Not regular | Regular |
| Not connected | Not connected | Weakly connected | Simply connected | Strongly connected | Strongly connected |

Fig. 9. Conceptual neighborhood diagram for transition rule B

The conceptual neighborhood diagram for transition rule C introduces several new transitions (figure 10). In particular, note that two region transitions are unidirectional: from regular, strongly connected to regular, simply connected and from regular, not connected to regular, simply connected. For example, changes at the boundary of a region can lead to a hole being closed up, but cannot create a hole in a regular, simply connected region without first going through an intermediate irregular (and strongly connected) stage.

| Not regular | Regular | Not regular | Regular | Not regular | Regular |
| Not connected | Not connected | Weakly connected | Simply connected | Strongly connected | Strongly connected |

Fig. 10. Conceptual neighborhood diagram for transition rule C

Finally, the conceptual neighborhood diagram for transition rule D enables almost all transitions to take place (figure 11). Although transition rule D represents our most unconstrained situation, there still remain some transitions that are not possible in a single step (such as regular, not connected to not regular, strongly connected).

Aside from providing concise pictorial summaries of the possible transitions for each transition rule, the conceptual neighborhood diagrams illustrate the range of different dynamic behaviors that can be achieved even with our rela-

| Not regular
Not connected | Regular
Not connected | Not regular
Weakly
connected | Regular
Simply
connected | Not regular
Strongly
connected | Regular
Strongly
connected |
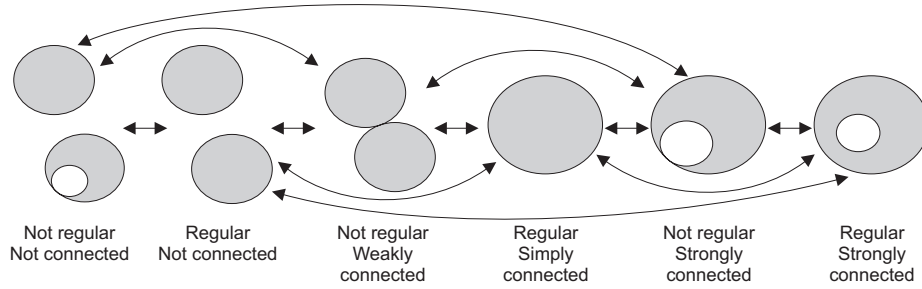
Fig. 11. Conceptual neighborhood diagram for transition rule D

tively simple formal model. As highlighted previously, an important aspect of our approach is that many of the transitions summarized in figures 1-4 can be detected using *local* rules. For example, transitions from regular to not regular simply require that some node be incident with more than two boundary edges, a condition that can be tested locally. As described previously, this knowledge can be used as the basis for efficient automated inferences about properties, such as connectedness, which may not be detectable using local algorithms.

## 6 Detecting events in dynamic regions

We now show how the formal model developed in this paper can be used to detect salient events in dynamic regions. A dynamic region is modeled as an ST scalar field $f : S \times T \rightarrow D$ (see section 4). In this section the domain D is simply the set $\{0, 1\}$. At any point in time t, we can assign each element δ of our triangulation Δ to a region R if some part of δ maps to the domain value 1 at that time, i.e., for all $\delta \in \Delta$, $\delta \in R$ if and only if there exists some $p \in S$ such that δ contains p and $f(p, t) = 1$.

In addition, when detecting changes in dynamic regions we may stipulate that, for the time interval under consideration, the evolution of R conforms to one of the transition rules defined in the previous section. Conforming with a particular transition rule requires that the dynamic region must be initialized in one of the valid states in that transition rule (shown in the conceptual neighborhood, section 5.2), and that all changes to the region are be performed in accordance with the triangle insertion and deletion methods for that transition rule (section 5). The following examples show how applying different transition rules enables different types of event to be detected.

6.1 Detection of movement and expansion

Applying transition rule A enables movement, stasis, expansion, and contraction to be modeled and detected. To recapitulate, these rules assume the initial region R is regular and only additions and deletions that result in regular regions are allowed. For example, figure 12 shows a sequence of stages from the development

of a dynamic region. The region is shown as the circular shape in figure 12, while the set of triangles where the region has been detected at a particular time is shown using shading.
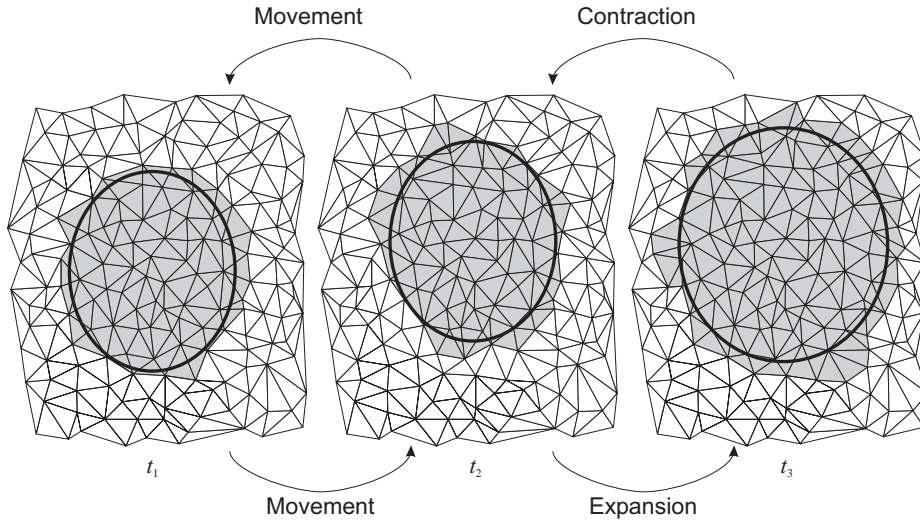

Fig. 12. Some events using boundary-regular transition rules

For any two times t and t' we define *expansion* as the event where $R(t) \subset R(t')$ and $t < t'$. Conversely, *contraction* is defined as the event where $R(t) \subset R(t')$ and $t > t'$. *Appearance* and *disappearance* can be defined of as special cases of expansion and contraction where $R(t) = \varnothing$. *Stasis* is defined as the event where $R(t) = R(t')$. *Movement* is defined as any transition between $R(t)$ and $R(t')$ that is not expansion, contraction, or stasis. Thus on figure 12 the transitions between successive stages of development have been labeled with the corresponding events.

Note that in the definitions above the time intervals t and t' do not need to be consecutive times at the finest temporal granularity available in our model. In figure 12, several intervening time steps have been omitted (between t_1 and t_2 and between t_2 and t_3). Using the definitions above, the size of the time interval chosen may determine what events are observed. At a coarser time interval the changes between t_1 and t_3 can be viewed simply as an expansion. Similarly, looking at time sub-intervals, such as changes that occurred between t_1 and t_2, may reveal different events. Thus, a more restrictive definition of *strong expansion* for two times t and t' would be as the event $R(u) \subset R(u')$ for all u, u' where $t \leq u < u' \leq t'$ (and similarly for *strong contraction*).

6.2 Detection of hole formation

Hole formation can be detected using transition rule B. Figure 13 shows an example of the detection of hole formation. Hole formation is defined as a change from a regular, simply connected region (at t_1) to a regular, connected region (at t_2). Hole disappearance is detected as the converse change. Note that the region may also be moving, expanding, or contracting without affecting the mechanism for detecting hole formation. In a similar way we could use transition rule B to detect the formation or disappearance of disconnected islands, occurring as a change from a regular, simply connected region to a regular, disconnected region.
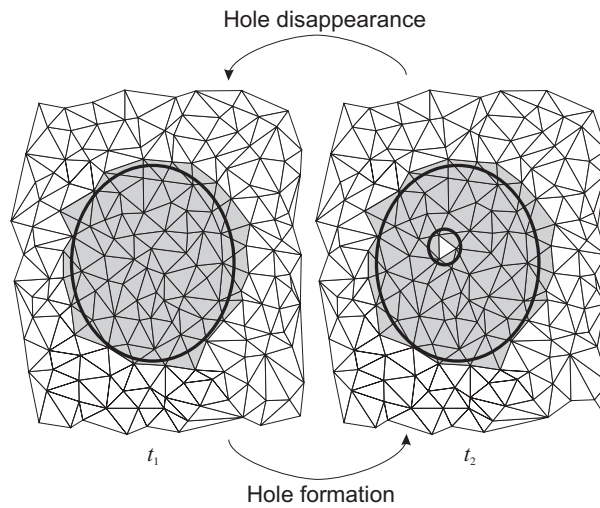

Fig. 13. Hole formation using general-regular transition rules

6.3 Detection of splitting

In order to detect splitting we need to use transition rule C. Recall that transition rule C allows the addition and removal of new triangles at the boundary of R irrespective of regularity. The development of a "pinch point" in figure 13 shows the process of *splitting*, defined as a change from a regular, connected to an irregular, weakly connected region. Splitting may be followed by *disaggregation*, defined as a change from an irregular, weakly connected to a regular, disconnected region. The converse processes of *aggregation* and *merging* can also be defined in a similar way.

6.4 Transition rule properties

Previous sections have illustrated how different transition rules can be used to detect different spatial events. Events can only be detected using the transition
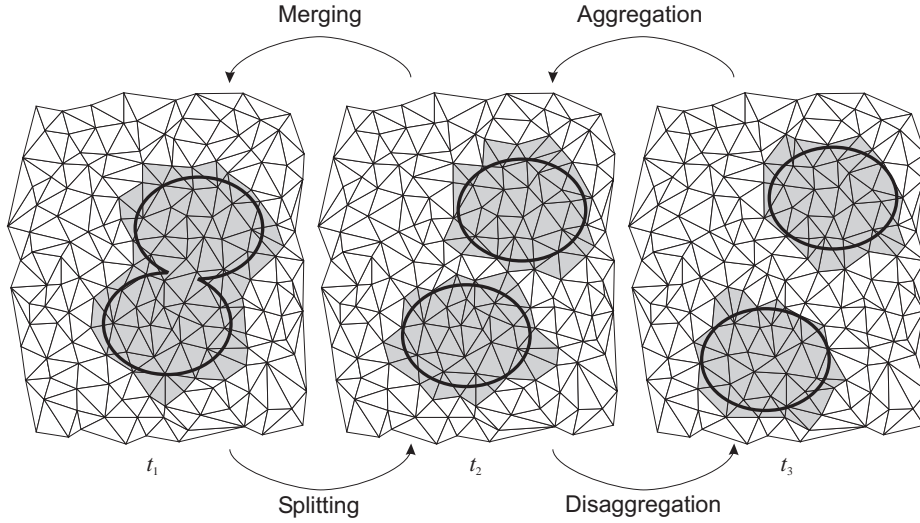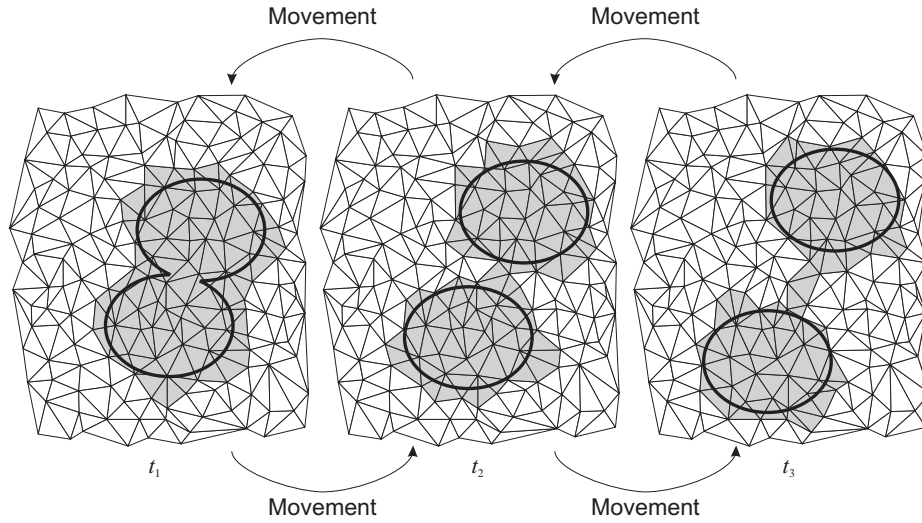
Fig. 14. Splitting and merging using boundary transition rules

rules for which they are defined. For example, even if the underlying continuous phenomena splits, this will not be detected using a region evolving through the application of transition rule A. Figure 15 shows the results of using transition rule A in such a situation. Because the transition rule A disallows transitions to irregular regions, the region in figure 15 will never reach the pinch point in 14. Further, since only boundary changes are permitted using transition rule A, the region can never progress directly to the regular, disconnected region (although this would be permitted using the transition rule B, for example).

Thus, at each time step we may ask for a particular transition rule, what proportion of the triangles in a dynamic region contain some part of the underlying dynamic field (i.e., errors of commission) or what proportion of the triangles in the triangulation that contain some part of the underlying dynamic field are part of the dynamic region (i.e., errors of omission). These proportions provide measures of the veracity of a particular transition rule. For phenomena that yield many splitting, hole, and island formation events, for example, we expect transition rule A to result in lower veracity (higher errors of omission and commission) than, say, transition rule D (see section 6.6 for further analysis).

Although some transition rules may provide less veracious representations of the underlying dynamic region they may also provide more useful inferences. For example, for a dynamic region based on transition rule C that was known to be connected at time t and has remained regular to a subsequent time t', we can infer that the R must still be connected at time t' (as discussed in previous sections, transition rule C prevents disconnection without an intermediate stage of irregularity). The same inference does not hold for transition rules B or D.

Fig. 15. Inability to detect splitting and merging using boundary-regular transition rules

6.5 Spatial and temporal granularity

In all the figures in this section, the triangulations have been somewhat exaggerated for clarity. In general we envisage triangulation frameworks that are much more dense, relative to the extent of the dynamic region under study. Thus, the triangulation mesh would be much finer and the edge effects visible in figures 12–15 would be minimized. This coarse spatial granularity is also associated with coarse temporal granularity in detecting events. For example, figure 14 contains significant spatial imprecision and associated edge effects. The example illustrates the detection of splitting in the underling dynamic region, but as a consequence of the coarse spatial granularity the splitting event is detected some time after the underlying dynamic regions have in actuality split. These "temporal edge effects" would also be greatly reduced by using a triangulation framework at a much finer spatial granularity.

6.6 Simulation

To empirically explore transition rule properties a simulation system was written in Java, in which the different transition rules were implemented tested against a variety of scenarios. One example scenario, which encapsulates several different types of dynamic events, is illustrated in Figure 16. The figure summarizes the following changes that occur to a dynamic field, $f : T \to S \to \{0, 1\}$ (where the oval region with the thick boundary indicates the initial state of the field at time t_0):

1. Hole appears in region

2. Hole in region grows
3. Region bursts (hole meets exterior of region)
4. New disconnected part of region appears
5. Disconnected part of region grows
6. Region merges
7. Region splits
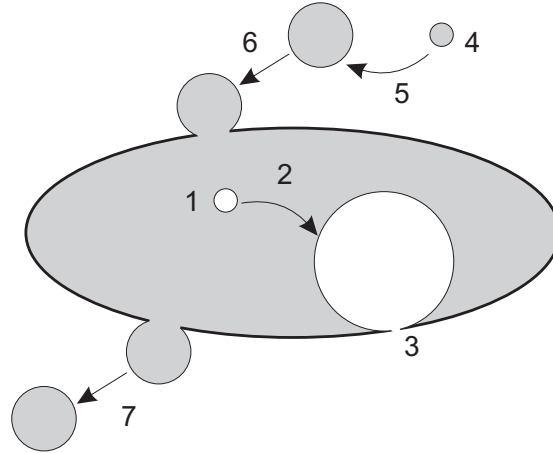
Fig. 16. Example dynamic field scenario

The simulation system allows the veracity of the transition rules to be tracked across the different events that occur. Figure 17 shows the levels of errors or omission and commission for each transition rule. The level of error is shown on the ordinate of each graph with time on the abscissa. The events from the example scenario above are also marked on each figure. The error levels are averaged over multiple simulation (10 runs) and a moving average was used to filter out fine detail and highlight the general trends.

The different graphs provide a "fingerprint" of events for each transition rule. For example, for transition rule A (Figure 17a, event 1), hole appearance cannot be detected and as a result, errors of commission increase as the hole grows (Figure 17a, event 2). Once the region "bursts" the changes can be detected at the boundary of the region, and errors of commission rapidly decrease (Figure 17a, event 3). Similarly, the appearance of new parts cannot be detected with transition rule A, so errors of omission increase as the new disconnected part of the region appears and grows (Figure 17a, events 4–5). Merging returns the region to a connected whole, reducing errors of omission (Figure 17a, event 6), followed by increasing errors of commission corresponding to the undetected splitting (Figure 17a, event 7).

Transition rule C (Figure 17c) follows a very similar pattern to transition rule A, except that the region splitting can be detected, so no errors of commission
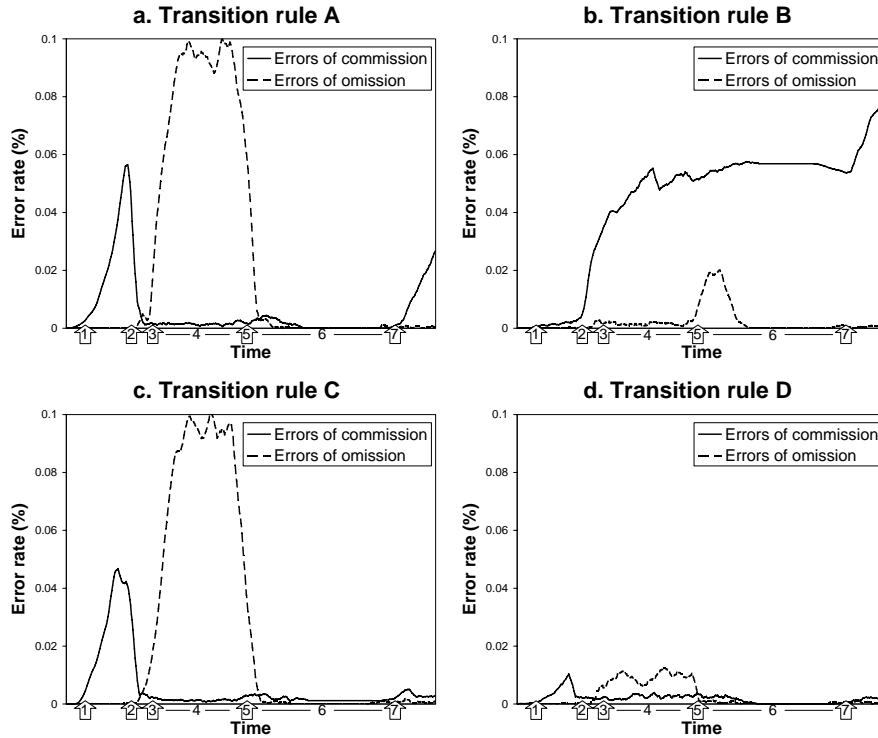
Fig. 17. Veracity of transition rules for example scenario in Figure 16

are associated with this event (Figure 17c, event 7). Transition rule B does adequately detect the hole formation (Figure 17b, events 1–2), but cannot detect the region bursting, leading to errors of commission following this event (Figure 17b, event 3). The merging and splitting events are not detected, leading to brief increases in the errors of omission and commission respectively (Figure 17b, events 6–7). As expected, the errors of omission and commission for transition rule D are very low for all events, and represent a control case for the data (these errors occur because of a slight delay in the simulation between an event occurring and the event being detected).

A variety of other scenarios and transition rule properties have been investigated using the simulation system. At this stage, these simulations are useful for guiding the formal model development. However, in future work further simulations are planned in the context of geosensor network application with both simulated and real field data.

7   Discussion and conclusion

This paper has set out a formal framework for modeling continuous spatial change in an event-oriented discrete spatiotemporal framework, and shown how this framework can be used for detecting spatial events, such as splitting, merging, movement, stasis, and hole formation and disappearance. The significance of this approach is that it provides a mechanism for the explicit representation of qualitative spatial events based on snapshot data, such as is generated by conventional sensor-based systems.

This paper provides a computational model for defining and detecting spatiotemporal events for a *single* region. The approach is being extended to spatial events involving *multiple* regions. There are a several options for achieving this. One option is for two non-empty dynamic regions $R_1, R_2 \in R(\Delta)$ is to consider the relationships between the dynamic region of overlap between regions at a particular time. We can then define for the two regions correspondences between this region of overlap and RCC8 or RCC5 (for example, $DR(R_1(t), R_2(t)) \equiv R_1(t) \cap R_2(t) = \varnothing$ and $\forall v \in \delta R_1(t).v \notin \delta R_2(t)$ where $R(t)$ is the extent of R at time t). Again, based on these relations we can derive the conceptual neighborhood for the region transitions, in turn using these to define events in the evolution of pairs of regions.

Other future work already highlighted in this paper includes:

– extensions for modeling continuous but not incremental evolution of regions;
– extensions for modeling uncertainty in regions, where it may be indeterminate whether or not a triangle is in a region, or uncertain to what extent a triangle is within a region; and
– further simulations, with both simulated and real field data.

While this paper has focused primarily on formal foundations, we end with a brief discussion of the target application area for our approach: geosensor networks. This application area has already been highlighted as important (section 2.2), and motivates several further directions for future research that are currently being investigated.

7.1   Geosensor network application

Section 1.1 introduced the example application of using a geosensor network for monitoring the dynamic spatial evolution of an oil slick. Such a geosensor network may be modeled as a triangulation in our formal model. If the absolute spatial location of each geosensor is known, for example using GPS, then this information may be used to triangulate continuous space. However, even where GPS information is unavailable (perhaps due to power constraints) other localization techniques may be used to derive the required qualitative spatial information to build the combinatorial map (such as acoustic range and direction finding, e.g., Chen et al., 2003). Triangles in the triangulation are bounded by three vertices (geosensors), with edges connecting pairs of geosensors. (Alternatively, it would also be possible to model each geosensor as a triangle in

our triangulation framework, with adjacency relationships to three other geosensors.) Each geosensor gathers information about the presence or absence of the oil slick in its immediate vicinity. Using this information, in combination with the dynamic triangulation transition rules described earlier, each geosensor could determine at a particular point in time whether the cells it bounds are part of a dynamic triangulation. In this way, the continuous movement of the oil slick could be tracked and spatial events detected.

Our triangulations may be envisaged as an additional "layer" in the geosensor network architecture, superimposed on top of the underlying communication network topology. However, current work is addressing the precise relationship between the communication network topology and the triangulation itself. While the model presented in this paper assumes a triangulation of the plane, it would be possible to generalize our formal model to allow any planar tessellation, not only triangulations, to be used as a spatial framework. Several of the major routing algorithms for geosensor networks (and mobile ad-hoc networks in general) generate network topologies that are planar graphs (e.g., Karp and Kung, 2000). Thus, it may be possible to integrate the communication network with the triangulation itself.

The information generated by the geosensors could be transmitted wirelessly to a central information processing system, using standard multi-hop communication transition rules (see Murthy and Manoj, 2004). However, the dynamic triangulation transition rules and the rules for detection of events are primarily local, and relate to changes in the neighborhood of a single triangle or geosensor. Current work is addressing the specification of *in-network* query algorithms, where spatial events are detected using local rules without the need for any centralized information processing or for queries that flood the entire geosensor network (Duckham et al., 2005).

Finally, the oil slick scenario assumes the geosensors are stationary and always switched on. In real geosensor networks sensors may be mobile, perhaps carried by wind or ocean currents, may switch themselves on or off, and may be added or removed from the network. Initial work to investigate the situation where nodes can become active or inactive within a static triangulation framework has already begun (Duckham et al., 2005). Planned research will also relax the assumption of a static spatial framework, using recent research into process-oriented approaches to geographic information as a basis (Worboys, 2005). Although the approach and formalism in this paper is different from the process algebra used in Worboys (2005), current work by the authors is aiming to integrate these two approaches (for example by encoding our four dynamic triangulation transition rules within a process algebra).

8   Acknowledgments

Bibliography

Abraham, T. and Roddick, J. (1999). Survey of spatio-temporal databases. *GeoInformatica*, 3:61–69.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114.

Braginsky, D. and Estrin, D. (2002). Rumor routing algorthim for sensor networks. In *Proc. 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 22–31.

Chen, J. C., Yip, L.and Elson, J., Wang, H., Maniezzo, D., Hudson, R. E., Yao, K., and Estrin, D. (2003). Coherent acoustic array processing and localization on wireless sensor network. *Proceedings of the IEEE*, 91(8).

Duckham, M., Nittel, S., and Worboys, M. (2005). Monitoring dynamic spatial fields using responsive geosensor networks. In *Proc. 13th ACM GIS 2005*. ACM Press. (Accepted, in press. Conference accepted 36% of submitted papers).

Dufourd, J.-F. and Puitg, F. (2000). Functional specification and prototyping with oriented combinatorial maps. *Comput. Geom. Theory Appl.*, 16(2):129–156.

Edmonds, J. (1960). A combinatorial representation for polyhedral surfaces. *Notices Amer. Math. Soc.*, 7:646.

Estrin, D., Govindan, R., and Heidemann, J. (2000). Embedding the Internet: Introduction. *Communications of the ACM*, 43(5):38–41.

Freksa, C. (1991). Conceptual neighborhood and its role in temporal and spatial reasoning. In Singh, M. and Travé-Massuyès, L., editors, *Decision Support Systems and Qualitative Reasoning*, pages 181–187, Amsterdam. North-Holland.

Galton, A. (2004). Fields and objects in space, time, and space-time. *Spatial Cognition and Computation*. forthcoming.

Grenon, P. and Smith, B. (2004). SNAP and SPAN: Towards dynamic spatial ontology. *Spatial Cognition and Computation*, 4(1):69–104.

Guibas, L. and Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of voronoi. *ACM Trans. Graph.*, 4(2):74–123.

Hornsby, K. and Egenhofer, M. J. (2000). Identity-based change: A foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, 14(3):207–224.

Kahn, J. M., Katz, R. H., and Pister, K. S. J. (1999). Next century challenges: Mobile networking for "smart dust". In *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 271–278.

Karp, B. and Kung, H. T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 243–254, Boston, MA. ACM Press.

Murthy, C. S. R. and Manoj, B. S. (2004). *Ad Hoc Wireless Networks : Architectures and Protocols*. Prentice Hall.

Nittel, S., Stefanidis, A., Cruz, I., Egenhofer, M., Goldin, D., Howard, A., Labrinidis, A., Madden, S., Voisard, A., and Worboys, M. (2004). Report from the First Workshop on Geo Sensor Networks. *ACM SIGMOD Record*, 33(1).

Perkins, C. E., Royer, E. M., Das, S. R., and Marina, M. K. (2001). Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications Magazine*, 8(1):16–28.

Singh, S., Woo, M., and Raghavendra, C. S. (1998). Power-aware routing in mobile ad hoc networks. In *Proc. 4th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 181–190.

Snodgrass, R., editor (1995). *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers.

Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., and D., E. (2004a). Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40.

Szewczyk, R., Polastre, J., Mainwaring, A., Anderson, J., and David Culler, D. (2004b). An analysis of a large-scale habitat monitoring application. In *Proc. 2nd ACM Conference on Embedded Networked Sensor Systems*, pages 214–226.

Tutte, W. (1984). *Graph Theory*. Cambridge University Press, Cambridge, England.

Worboys, M. (1994). A unified model of spatial and temporal information. *Computer Journal*, 37(1):26–34.

Worboys, M. and Hornsby, K. (2004). From objects to events: GEM, the geospatial event model. In Egenhofer, M., Freksa, C., and Miller, H., editors, *Third International Conference on GIScience*, volume 3234 of *Lecture Notes in Computer Science*, pages 327–344. Berlin: Springer Verlag.

Worboys, M. F. (2005). Event-oriented aproaches to geographic phenomena. *International Journal of Geographic Information Science*. In press.

Worboys, M. F. and Bofakos, P. (1993). A canonical model for a class of areal spatial data. In Abel, D. and Ooi, B., editors, *Proceedings of the 3rd International Symposium, SSD93*, Lecture Notes in Computer Science, pages 36–52. Berlin: Springer Verlag.

Xu, Y. and Lee, W.-C. (2005). Window query processing in highly dynamic geosensor networks: Issues and solutions. In Stefanidis, A. and Nittel, S., editors, *Geosensor networks*, pages 31–52. CRC Press, Boca Raton, FL.