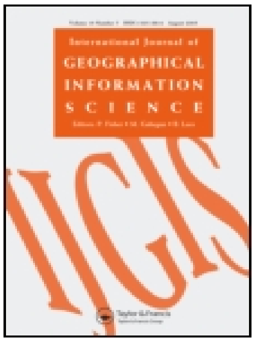

International Journal of Geographical Information Systems

ISSN: 0269-3798 (Print) (Online) Journal homepage: http://www.tandfonline.com/loi/tgis19

An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data

Donna J. Peuquet & Niu Duan

To cite this article: Donna J. Peuquet & Niu Duan (1995) An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data, International Journal of Geographical Information Systems, 9:1, 7-24, DOI: 10.1080/02693799508902022

To link to this article: http://dx.doi.org/10.1080/02693799508902022

Published online: 05 Feb 2007.

INT. J. GEOGRAPHICAL INFORMATION SYSTEMS, 1995, VOL. 9, NO. 1, 7–24

Research Article

An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data

DONNA J. PEUQUET

Department of Geography and Earth Systems Science Center, The Pennsylvania State University, University Park, PA 16802, U.S.A.

and NIU DUAN

Department of Industrial and Management Systems Engineering, The Pennsylvania State University, University Park, PA 16802, U.S.A.

(*Received 21 April 1993; accepted 10 February 1994*)

Abstract Representations historically used within GIS assume a world that exists only in the present. Information contained within a spatial database may be added-to or modified over time, but a sense of change or dynamics through time is not maintained. This limitation of current GIS capabilities has recently received substantial attention, given the increasingly urgent need to better understand geographical processes and the cause-and-effect interrelationships between human activities and the environment. Models proposed so-far for the representation of spatiotemporal data are extensions of traditional raster and vector representations that can be seen as location- or feature-based, respectively, and are therefore best organized for performing either location-based or feature-based queries. Neither form is as well-suited for analysing overall temporal relationships of events and patterns of events throughout a geographical area as a temporally-based representation.

In the current paper, a new spatio-temporal data model is proposed that is based on *time* as its organizational basis, and is thereby intended to facilitate analysis of temporal relationships and patterns of change through time. This model is named the Event-based Spatio Temporal Data Model (ESTDM). It is shown that temporally-based queries relating to locations can be implemented in an efficient and conceptually straightforward manner using ESTDM by describing algorithms for three fundamental temporally-based retrieval tasks based on this model: (1) retrieving location(s) that changed to a given value at a given time, (2) retrieving location(s) that changed to a given value over a given temporal interval, and (3) calculation of the total area that has changed to a given value over a given temporal interval. An empirical comparison of the space efficiency of ESTDM and compressed and uncompressed forms of the 'snapshot' model is also given, showing that ESTDM is also a compact representation of spatio-temporal information.

1. Introduction

The need to better understand the effects of man's activities on the natural environment at all geographic scales is now viewed with increasing urgency. In natural resource management within the developed world, the emphasis is shifting from inventory and exploitation toward maintaining the long-term productivity of the environment. This task requires integrated and broad-scale process analysis in order to better understand natural and human processes and how they are interrelated. Global Circulation Models (GCMs) are currently being used to study climate dynamics, ocean dynamics and global warming (Houghton *et al.* 1990). The need for a more detailed examination and understanding of man-environment interactions at urban and regional scales is also a continuing priority. All of these require the analysis of *change* through time and of *patterns* of change through time. The advent of remotely-sensed satellite data in addition to the accumulation of other spatio-temporal observational data has made the empirical study of large-scale, complex spatio-temporal processes possible, further increasing the demand for integrated computer-based tools for this task. As a result of such data storage and processing needs, enhancement of the spatio-temporal capabilities of geographical information systems is now an issue that is receiving attention. A key element in this work is how to represent geographical phenomena in time, as well as in space, so that spatio-temporal data can be effectively stored and analysed.

In the current paper, a new type of spatio-temporal data model is proposed which is based on *time* as its organizational basis and thereby is intended to facilitate analysis of temporal relationships and patterns of change through time. The specific spatio-temporal data model described is called ESTDM (the Event-based Spatio Temporal Data Model). This data model also represents one component of the Triad Database Framework as originally proposed by Peuquet (1994).

In the remainder of this paper, current and potential approaches for representing spatio-temporal data will be discussed within the context of varying tasks and types of queries. The ESTDM will subsequently be defined and its advantages for temporal analysis discussed. Algorithms for performing some basic tasks using ESTDM will then be described. An empirical examination of the storage efficiency of ESTDM will also be given. Finally, potential variations of this data model and areas for future research will be discussed.

2. Form versus function for spatio-temporal analysis

2.1. *Current approaches*

Any representational scheme is inextricably linked with a set of specific uses. This fact was demonstrated within the spatial realm during the lengthy raster versus vector debate that began in the 1970s (Peuquet 1978). Functional trade offs that have become recognized between these two representational approaches and with other non-spatially oriented approaches, Relational Database Management Systems in particular, have resulted in modern GISs being implemented increasingly on the basis of a multi-representational database design. If GISs in the future are to provide sophisticated temporal analysis capabilities as well as the ability to efficiently answer specific types of time-based queries, it is necessary to utilize a type of representation which is specifically suited to that type of application.

The only data model available within existing GIS that can be viewed as a spatio-temporal data model is a series of 'snapshot' images, as shown graphically in figure 1. This type of representation simply employs the grid data model, using a sequence of spatially-registered grids. Instead of a single gridded file representing a complete thematic map layer as in a static (i.e., non-temporal) spatial database, each gridded image, S_i, represents a 'world state' relative to a given thematic domain, storing a complete image, or 'snapshot', at a known point in time, t_i. Each cell within a separate snapshot contains the value for the corresponding location at that time. This approach is conceptually straightforward, and the 'world state' for any given point in time within the recorded temporal interval can be easily retrieved or interpolated. Nevertheless, the actual changes that occurred at locations between given points in time are not explicitly

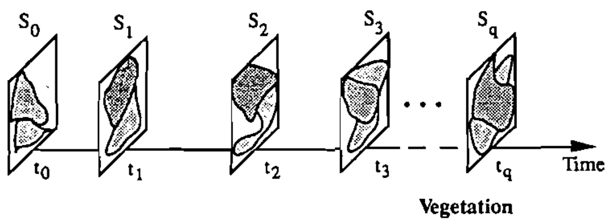

Figure 1. The Snapshot Approach. Note that the temporal distance between 'snapshots' is not necessarily uniform. Here, S_i denotes a given snapshot representing the 'world state' at time t_i.

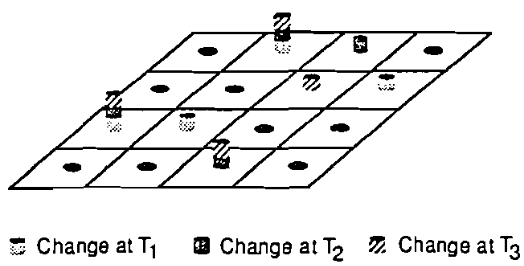

Figure 2. A temporal grid with a variable-length list attached to each grid cell denoting successive changes (from Langran 1992).

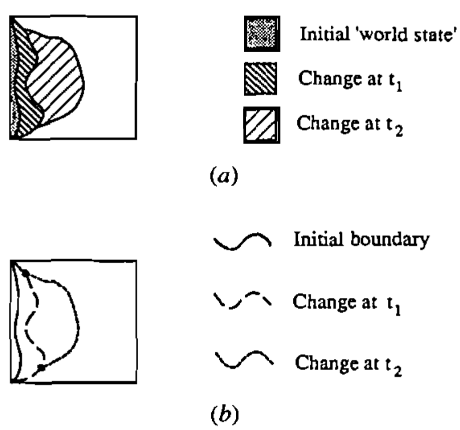

Figure 3. The Amendment Vector Approach, showing urban encroachment (after Langran 1992). (*a*) Temporal composite of areal changes. (*b*) Temporal composite of areal changes noting only amendment vectors (i.e., boundary changes).

stored, and can only be derived by comparing the pixel value differences between successive snapshots.

Another potential approach based on the grid model, which was proposed within a GIS context by Langran (1990) and is also used in electronic circuitry design analysis (Fujimoto 1990), is to represent each pixel in the gridded array of discretized locations as a list (c.f., figure 2). With this representation, a change at a given location is added to the beginning of the list for that location. The result is a set of variable-length lists referenced to grid cells. The 'present' (i.e., most recently recorded) world state for the entire area is easily retrieved as the first value stored in all of the locationally-referenced lists. This representation has the advantage of storing only the changes related to specific locations, and has the related additional advantage of avoiding the storage of redundant information (i.e., values for locations which remain unchanged) in contrast with the snapshot representation.

Several spatio-temporal models have recently been proposed that explicitly record spatial changes through time as they relate to specific geographic/cartographic entities instead of locations (Hazelton 1991, Kelmelis 1991, Langran 1992). At a broad conceptual level, all of these proposed models represent extensions of the classical vector representational approach. These spatio-temporal models rely on the concept of amendments, where any changes subsequent to some initial point in time in the configuration of polygonal or liner entities are incrementally recorded as additions to the original entities. The first of these models was proposed by Langran (1989 b) and relies on what she described as 'ammendment vectors'. As a simple graphic example, figure 3 shows the areal expansion of a single urbanized area at specific points in time from an initial state at time t_0 through time t_2. These successive increments, shown as a single, temporal composite map as in figure 3 (*a*) are stored in Langran's proposed model by noting only the 'amendment vectors', as in figure 3 (*b*). The time that any change occurred is recorded as an attribute of each amendment. This organization allows the integrity of individual features (e.g., lakes, roads, etc.), components of those features (e.g., boundary lines) and their spatial interrelationships to be explicitly maintained over time.

Hazelton utilized this basic idea within a 4-dimensional space-time Cartesian space, and proposed an extended hierarchy comprised of nodes, lines, polygons, polyhedra, polytopes and polytope families as the conceptual organizational basis (Hazelton, *et al.* 1990). Each of these, in succession, adds a single dimension so that polyhedra are 3-dimensional enclosed areas, polytopes are 4-dimensional enclosed areas, etc. Kelmelis proposed a similar feature hierarchy of nodes, lines surfaces and volumes within a 4-dimensional space-time Cartesian coordinate space as an extension of the DLG-3 data model (Guptill 1990, Kelmelis 1991).

All of the extended grid and vector models described above incorporate the temporal dimension in some way while still retaining their fundamental organizational basis. They thereby also retain their relative functional advantages and disadvantages. All vector models are feature-based in the sense that all locational and temporal information, as well as other types of attribute information, is stored *relative* to specific geographic/cartographic features, and/or the topologically-defined elements (lines and nodes) which make up those features (Peuquet 1984). In other words, geographic/cartographic entities serve as the basic conceptual element and organizational basis of the representation. Conversely, the grid model, or more generally any tessellation model, can be said to be location-based, since all other information is stored relative to specific locations.

Because of these two fundamentally different orderings of stored information, a vector model can be used most effectively to store information and perform tasks relative to spatial features, including topological relationships between them over space, but not nearly as effectively to store information and perform tasks relating to a specific location or set of locations. Conversely, a grid or tessellation model can much more effectively perform tasks relative to specific locations and locational overlay, i.e., locational set relationships (Peuquet 1988).

Associating additional temporal information with individual features then allows changes through time to individual features or their components to be easily traced and compared, including changes in their spatial topological relationships. Similarly, associating temporal information with locations allows the history of individual locations and sets of locations to be traced and compared. Locational overlay operations can also be used in a location-based spatio-temporal representation to characterize locations on the basis of the co-location of multiple changes or types of change.

In using a spatio-temporal GIS to analyse processes, it is also essential to be able to examine change on the basis of time, retrieving locations and features on the basis of the temporal relationships of a specified event and, moreover, to be able to examine overall patterns of temporal relationships. These patterns can potentially be very complex. In a similar manner that a spatial distribution pattern can be random, uniform or clustered, a temporal distribution can be chaotic, steady state or cyclic. Change through time can also be converging, diverging or combinations such as a dampened oscillation. Individual events can be characterized as clustered, forming 'episodes' that perhaps can be further grouped into 'cycles'. Perfectly cyclical distributions is an important form of steady-state behaviour over longer temporal intervals. Thus, a series of dry years in California can be grouped as a drought episode. This drought episode may in turn be part of the El Niño cycle. Variations in the length of El Niño cycles may be seen as chaotic, as may be governmental response to various economic and social repercussions of El Niño.

Specific examples of queries which relate geographical changes occurring with respect to specific *temporal relationships* would include:

— When did the last forest fire occur anywhere in the area?
— What areas are known to have suffered a landslide within one week of a major storm event?
— Which census tracts increased in unemployment within one year after closing of the local navy base?

Thus in addition to extending traditional raster-based and vector-based approaches in order to incorporate temporal change, change also needs to be explicitly modelled as a function of time. The need for a time-based representation for examining temporal relationships and patterns was also described by Langran and Chrisman (1988), but no example of such a representation was given.

2.2. *The time-based approach*

In the time-based approach shown graphically in figure 4, location in time becomes the primary organizational basis for recording change. The sequence of events through time, representing the spatio-temporal manifestation of some process, is noted via a time-line or temporal vector, i.e., a line through the single dimension of time instead of a two-dimensional surface over space. This time-line is shown in figure 4 as the 'event list'. Such a time-line, then, represents an ordered progression through time of known

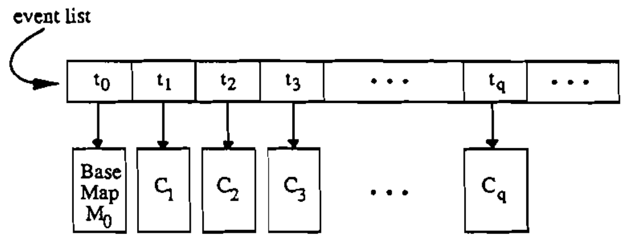

Figure 4. Representation of change organized as a function of time. Here, t_i contains a time value (e.g., day, month and year) and C_i contains all changes that occurred at time t_i.

changes from some known starting date or moment to some other known, later, point in time. Each location in time along the time-line (with its temporal location noted as $t_0, t_1 \ldots, t_n$) can have associated with it a particular set of locations and features in space-time that changed (or were observed as having changed) at that particular time and the new value to which they changed.

Recording only times when change occurs as opposed to a complete time-line that contains an entry for every 'tick of the clock' at a given temporal resolution can also be viewed as temporal run-length-encoding. This is analogous to raster run-length encoding for recording spatial variation. In raster run-length encoding, a value is recorded only when the value is different than the last one encountered along that scan line. The length of the run of contiguous y-locations along the same or scan-line with that same value is also recorded either as starting and ending y-locations or occasionally simply as an increment value, Δy. Thus it can be said for run-length encoding generally, only 'meaningful' locations in time or space are recorded. In the spatial domain, this is usually coincident with some change over space. This can be an edge, such as a municipal boundary or the change in vegetation from forest to meadow at a set of locations.

The top-level of information is called the 'event list' since an event usually represents a change in state (i.e., change in some property, or attribute, value) that can be denoted as such for some feature, location or set of features or locations. In other words, the *changes* relating to *times* are explicitly stored, and only the changes.

Besides sudden change as might be caused by some catastrophic event such as a forest fire or industrial plant closing, change can also be gradual, such as the amount of rainfall or income level associated with a particular x,y location, drainage basin, county, etc. For such instances of gradual change, a change 'event' is recorded at the time when the amount of accumulated change since the last recorded change is considered to be significant or by some other domain-specific rule. The problem of determining which location in time to store when casting gradual or continuous change through time into a discrete framework (as is normally necessary for any computer-based application) is the temporal equivalent of the classical, and multifaceted, spatial grid-cell sampling problem (Burrough 1986, Goodchild, *et al.* 1992) and is thus an issue for continuing research in spatio-temporal representations.

In the event-based approach, the time associated with each change (i.e., each event) is stored in increasing order from the initial 'world state' at time t_0 (e.g., 21 January 1952) to the latest recorded change at time t_n. These may be recorded at any desired temporal resolution. For most phenomena, the length of any temporal interval (i.e., temporal distance between t_{i-1} and t_i), and any other such interval will be unequal. Associated with each t_i are the changes which occurred between t_{i-1} and t_i. The only exception to this that the entire 'base map' or starting world state must be stored with the first recorded time, t_0. The changes associated with any t_i may also be extensive, affecting a large geographical area and many features within it, or perhaps may be only a single location or feature.

3. A time-based spatial data model

3.1. *General description*

Based on the discussion above, a specific data model is now defined. The proposed data model, which is called the Event-based SpatioTemporal Data Model (ESTDM), represents a specific example of the time-based approach that temporally orders changes to *locations* within a pre-specified geographical area. A generalized form of ESTDM will first be described before presenting the details as actually implemented.

ESTDM in its simplest form stores specific changes associated with each time, some t_i, in the event list as shown in figure 5. The specific stored temporal location (i.e., t_i) is called a *time-stamp*, using the convention from Temporal Database Management Systems (DBMS) terminology (Jensen *et al.* 1993). It is assumed that each event list and associated changes relate to a single thematic domain (e.g., land use of population). The set of changes, C_i, recorded for any t_i, as shown in figure 4, consists of the set of each x,y location which changed since t_{i-1}, and the new value, v, that each such location exhibits at t.

One obvious potential disadvantage of this method for storing changes is that the number of x,y,v triplets to be stored is directly related to the amount of area (i.e., the total number of discrete locations) which changed between t_i and t_{i-1}. In the ESTDM, this effect is minimized, as can be seen diagrammatically in figure 6. The representation of changes for a specific time consists of, first, grouping together individual cell locations, i.e., x, y pairs, which share a common new value, v. Such a value-specific group is stored within a single sub-structure we call a *component* (c.f., figure 6). In other words, all locations that have changed to the same values within a single thematic layer,

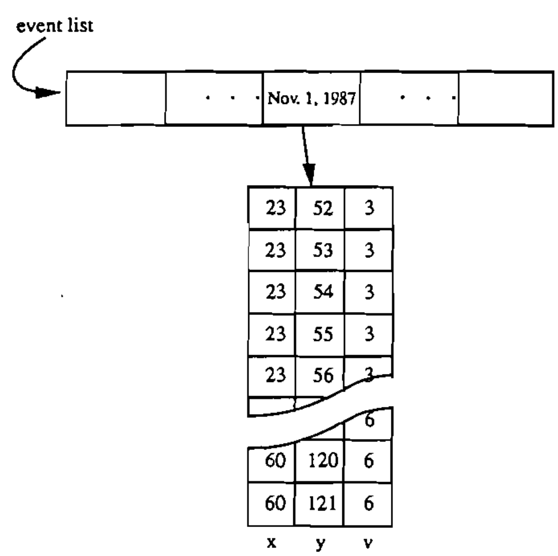

Figure 5. Detail of figure 4 showing the contents of C_i stored as individual x, y locations that changed with the new value, v.

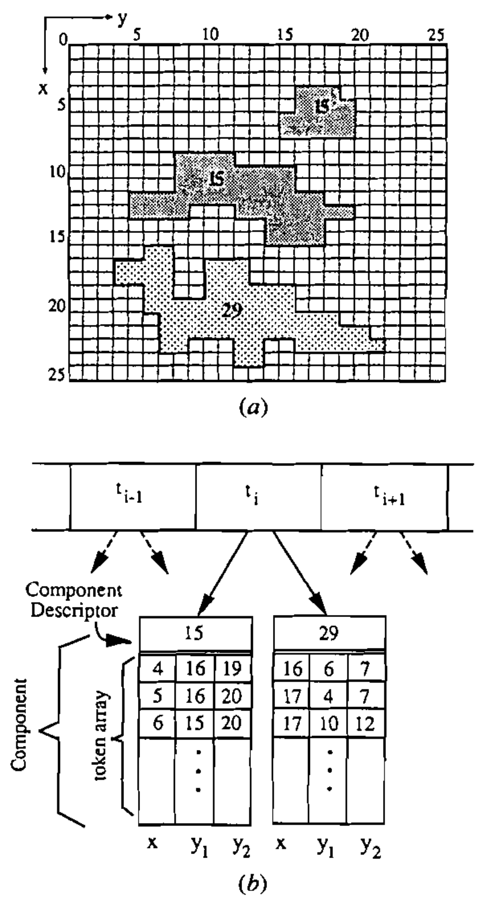

Figure 6. Spatial changes at tme t_i displayed as a simplified map (a) and the corresponding event components (b).

regardless of their location or previous value, are stored together as members of the same component. This means that any given value is stored only once per event instead of once for every x, y location, and that a separate component structure is stored for each value to which at least one locational cell has changed. By storing locations in this manner, it is also possible to apply raster run-length-encoding within each component in order to reduce the volume of storage space required for recording locations. Grouping locations which changed to a common value also greatly facilitates certain search tasks, as will be discussed later in this paper.

As seen in figure 6, the component structure is defined as having two primary elements: the new value, called a component descriptor, and an array of locational elements we call *tokens*. A single token represents a set of consecutive cells along a row in a griddled map utilizing run-length encoding. It consists of three entries: the row number, x, the first column number, y_1, and last column number, y_2 where y_1 indicates the left-most column and y_2 indicates the right-most column of the same row x of contiguous cells with the same value, v. Thus, the simple map in figure 6 could represent, for example, land use *changes* between t_i and t_{i-1} (locations that remained unchanged during this interval are ignored). Since all changes over the entire area were changes to only two different values, only two components result at t_i. One component contains all locations with changed to 'value 15' (e.g., pine forest) and the second denoting locations which changed to 'value 29' (e.g., barren land).

Utilizing the neutral terms of *components* and *tokens*, the event-based model can be defined in general as being composed of a series or list or temporal locations, each location corresponding to a single point in a 1-D time continuum, and one or more components associated with each of these points containing the new values and the elements to which they pertain. This becomes helpful in the context of potential variations to the initial structure, as those which will be described later in this paper.

3.2. *The ESTDM data structure*

The detailed ESTDM data structure for representing changing locations can now be described. The ESTDM structure, as implemented, consists of a header, a base map that defines the initial world state for the entire geographical area at t_0, and an event list with set of components attached to individual event entries in the event list (see figure 7). The definition of ESTDM as implemented in the C Programming Language is given in the Appendix. A single ESTDM-formatted file that represents the spatio-temporal dynamics of a single thematic domain for a specific geographic area, equivalent to a single thematic map layer, is called an *EST series* (i.e., an Event-based SpatioTemporal series).

The header contains the name of the thematic domain a pointer to the base map, the name of the base map, the time-stamp of the initial time value associated with the base map, and pointers to the first and last elements of the event list. The base map consists of a complete raster run-length-encoded 'snapshot' image of the entire geographical area represented. Each event entry in the event list contains a time-stamp, a list of pointers pointing to each event component, and a pair of pointers, *prev* and *next*, that point to the previous and the next element in the event list, respectively. The pointer *prev* of the first element points back to the header and the pointer *next* of the last element in the event list is assigned the value **NULL**. The event list is thus constructed as a doubly-linked list.

The use of pointers to connect adjacent entries in the event list allows new events that happen as time progresses forward to be easily added. This conceptually entails simply adding new events to the end of the event list because of its temporal ordering. At the implementation level, the use of pointers allows ESTDM-formatted files (i.e., EST series) to expand in order to accommodate the additional data while avoiding the need to physically recopy the file. Adding a new event (which has occurred at some

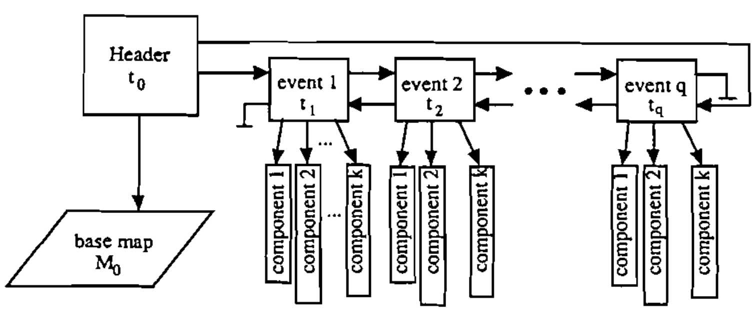

Figure 7. The ESTDM showing all primary elements and the pointer structure.

time more recent than the last event already in the event list) onto the end of the event list requires only the following straightforward pointer adjustments:

1. Change pointer *next* of the last entry in the event list from **NULL** to the storage location of the new event (i.e., the timestamp with its associated components)
2. Assign the location of the last entry in the event list as the value of pointer *prev* of the new event.
3. Assign the pointer *next* of the new event the value **NULL**.

The new event thereby becomes the last event in the updated event list.

The use of backward pointers allows the event list to be searched in reverse order as well as forward order. The separation of the base map in an EST series as a separate element and not as event components is done for both representational and efficiency reasons. It is obvious that t_0 must be considered 'the beginning of time' as far as the data are concerned. This means that the nature of the values for all locations at that time are unique in that it would not be valid to view these initial locational values as 'changes' from some previous time since there is no previous time. It is also because of this that the pointer *prev* of the first event is **NULL** in the implementation described here to allow algorithmic clarity. On a practical level, this initial world state is the only necessarily exhaustive inventory of values for all locations, and thereby is used as the base from which any later world state can be derived and reformatted into a complete snapshot, if desired, by accumulating subsequent changes for the entire area in sequence to the desired time. For data representing temporally transient phenomena associated with specific types of events, the base map may also not be relevant, such as for storm events, since there is no temporal continuity relative to a specific geographical area.

4. Functional evaluation of ESTDM

The most significant and unique capabilities of ESTDM in a GIS context arise from the ability to perform temporal manipulations on the data (e.g., temporal scale change) and temporally-based comparisons in a sequential manner. The primary ordering of information on the basis of time facilitates search and retrieval of not only specific temporal intervals looking in either a forward of backward temporal direction, but also of change to specific values within those temporal intervals. Such an ordering also makes comparison of different temporal sequences for differing thematic domains, or of the same thematic domain in different geographical areas, a straightforward task of comparing two or more EST series by comparing the timestamps in their respective event lists, in sequence. Temporal sequences within the same EST series may also easily be compared in a similar manner.

The hierarchical organization of data within the ESTDM offers additional functional advantages. For comparing *only* the times at which events occur (looking for overall temporal pattern), the times alone are retrieved directly from the event list without need of retrieving the associated values or locations. Also, since changes associated with each event are organized on the basis of each unique value occurring in association with that event, the frequency and variability of occurrences of specific values, regardless of their spatial location, can easily be examined as change progresses through time. Using the third level in the storage organization, questions relating to locational change (e.g, determination of which locations changed to a specific value or set of values at a given time) can be easily answered as a more complete retrieval operation.

4.1. *Temporally based queries*

Specific algorithms for performing several elemental temporally-based queries are given below. Each of these algorithms, as described, deal only with a single EST series. Algorithms for performing similar queries relating two or more series can be derived as extensions of these. The following algorithms are given simply to serve as examples of how implementing time-based queries can be straightforward and efficient using ESTDM. These would also serve as some of the elemental building-blocks for implementing more complex and application-specific tasks:

1. Retrieve all locations which changed to a given value at a given time.
2. Retrieve all locations which changed between t_a and t_b with their new values.
3. Calculate the total areal change between t_a and t_b.

4.1.1. *Retrieve location(s) which changed to a given value at a given time*

The retrieval of all locations which changed to a particular given value, gv, at time t_i is the most fundamental retrieval task for which ESTDM is particularly suited. The basic procedure for accomplishing this task using ESTDM is a two-stage search. The first stage of the search is to find the event with the desired time-stamp within the event list. The second stage is to find the component, c, associated with that event whose descriptor matches the given value, v. All locations within that component are then returned. Since the event list is arranged in increasing temporal order, the first stage of the search relies on a comparison of whether the given time, gt, is greater than the time-stamp of the current event in the event list. If $t_{first} > gt$, then the entire list occurred after the desired time and the search returns **NULL**. Otherwise, the search continues until $t_e \geq gt$, where t_e is the time associated with event e. Here, we assume that the value of gt does not necessarily match any time-stamp stored in the event list. If $t_e \neq gt$, we utilize the simple rule of closest temporal distance as to whether t_e or t_{e-1} is selected, although this can change depending on the application.

Remembering that EST is defined as a complex structure, we can more formally describe the general logic of the algorithm as follows:

```
Procedure Get_ClocsVal (EST,gt,gv)
begin
    if (t_first > gt) return NULL;
    foreach event in EST
        if (t_e >= gt)
            begin
                if (gt <= ((t_e + t_e-1)/2)) event = previous(event);
                foreach c of event
                    if (c(v) = = gv) return c(xylist);
            end;
    return NULL;
end;
```

Since both the search of the event list and the search of component descriptors within the desired event once found are exhaustive, linear searches in the algorithm as described above, the worst-case efficiency is simply $O(ne + nc)$, where ne = the total number of events in the event list and nc = the maximum number of components for any given event. This can be improved to $O((\log ne) + nc)$ by using any $O(\log n)$ search, where n denotes the *total* number of elements to be searched (Knuth 1973).

Performing the same task utlizing the snapshot model would require the following three steps: 1. Find the map with the right time-stamp in the map sequence, 2. Create a difference map between that map and the preceding map. This difference map would then contain the new values in all cells whose values had changed from the preceding map and zero or NULL in all cells whose values had not changed. 3. Find those cells whose contents match the given property changes. The first and third steps are generally equivalent to the two phase search as in the ESTDM-based algorithm above, the primary difference, however, is the addition of the second step to create the difference map. This process is necessarily exhaustive, always requiring ($nx*ny$) cell-by-cell comparisons between two adjacent snapshots, where n = the total number of cells. This means that the entire task is performed in O(n) time for a complete snapshot image.

4.1.2. *Retrieve all Locations which Changed to a Given Value Over a Given Temporal Interval*

This procedure is a simple variation on the previous one that utilizes a range of temporal values at the first level of search, retrieving components for all events from a starting time of *gts* to an ending (i.e., finishing) time of *gtf*. For the sake of simplicity, we assume here that the temporal distance between *gts* and *gtf* is wide enough that at least one event will be found.

Procedure Get_ClocsInt (EST,gts,gtf,gv)
begin
  if ($t_{first} > gt$) return NULL;
  foreach event in EST
    if ($gts \leq t_e \leq gf$)
      foreach c of event
        if ($c(v)$ = = gv) return c(xylist);
  return NULL;
end;

From the above, it can be seen that this task has the same logical structure that involves the same two-level search as the previous algorithm. Given retrieval of events with a temporal range of *gtf-gts* instead of a single event from a temporally-ordered list, the time efficiency would be O((log ne) + ($nf*nc$)), where (log ne) is the amount of time needed to search the event list for the starting event. Since all events after the starting event to the finishing event, *nf*, are then retrieved sequentially, this is *nf* additional steps. For each event, *nc* additional steps are required to examine each component for each event. This means that after the starting event is found, ($nf*nc$) steps are required. The worst-case in terms of efficiency would be the case where the starting time coincides with the first event in the event list and the finishing time coincides with the last event in the event list. This would require all components for all events in the event list to be examined. The resulting efficiency would be O($ne*nc$).

4.1.3. *Calculate total change in area to a given value over a given temporal interval*

Finding the amount of areal change is another basic spatio-temporal query, both for finding how much change has taken place over a specific temporal duration, but also for calculating the rate of change over that temporal interval. Within the ESTDM, the amount of areal change over a previously defined temporal interval is represented by the total number of areal units represented within a component or within the c(xylist) returned by either of the algorithms above. Since run-length coding is used for both of these, counting the total number of areal units requires a very simple procedure that accounts for the cells not explicitly noted in the structure. If we assume the input, xylist, represents a run-length-encoded list of x,y cell locations that is returned from either of the algorithms above, an ESTDM areal change algorithm reduces to a simple counting procedure as follows:

```
Procedure area (xylist)
begin
    area = 0;
    foreach token (x, y1, y2), of xylist
        area = area + ( y2 - y1 + 1);
    return area;
end;
```

Obviously, the time efficiency of this procedure is a direct linear function of the number of compacted records in the xylist.

4.2. *Storage efficiency of ESTDM*

ESTDM is also found to be very space efficient. Although the purpose of the ESTDM is to facilitate specific types of queries and thus is not intended to supplant other spatio-temporal data modelling approaches, it is compared here with the snapshot data model in order to provide a general notion of its level of storage efficiency. The snapshot model is 'comparable' in the sense that both contain the same information elements of time, location and value, although ordered differently. Storage efficiency for ESTDM or any other time-based model becomes a particularly important issue if such a representation is to be only one component of a multi-representational database design scheme.

The worst case for both the ESTDM and snapshot data models in terms of storage compactness occurs when the stored data exhibits maximum temporal as well as maximum spatial variability. This can be described as the situation where every cell (i.e., every location over the whole area) is different from its adjacent neighbors in space and also the value for any cell at time t_i is different from the value for that cell at times t_{i-1} and t_{i+1}. This worst case in terms of a spatio-temporal distribution would be a checkerboard pattern where the values at all locations alternate or progressively change at every recorded time. The storage efficiency of the snapshot model and ESTDM are equivalent in such a worst-case. This is easily determined as follows.

Assuming no compression, the snapshot model requires storage of $(x*y)$ values; one for each cell for any given t_i. Since all values for the entire image are always recorded for each t_i included in the snapshot representation whether a location changed in value or not, the total number of values to be stored is $(x*y*t)$. If x, y and t can be considered equivalent as individual orthagonal axes in three dimensions, then the worst-case space efficiency for the snapshot model can be stated as $O(n^3)$. For ESTDM, $(x*y)$ elements must also be stored for each t_i if the new value for each x, y location has changed since t_{i-1} and no two adjacent cells have the same value. If this occurs for each successive t, then the total storage required is also $(x*y*t)$, or $O(n^3)$. Fortunately, it seems safe to assume that the worst-case, alternating checkerboard distribution would in reality be extremely rare, to say the least, particularly for natural phenomena,

If raster run-length encoding can be used for compaction in the snapshot model, the number of elements that need to be stored for any individual snapshot would then decrease as the size of contiguous areas with the same value at a given t_i increase.

This is also the case with ESTDM. With the snapshot model, however, values for the entire image must be stored, regardless of how many or how few cells actually changed in value since the preceding snapshot at t_{i-1}. Furthermore, if the snapshot model is to be used for studying temporal *change*, any form of compression used on individual snapshots would significantly hamper the ability to compare snapshots on a location-by-location basis; a task which is necessary for deriving changes from the snapshot representation as a preliminary step.

Utilizing ESTDM, *only* the changes are recorded. With run-length encoded value components used in the full ESTDM, the amount of storage required is affected by the number of different values that occur over the entire area (determining the number of components for any given t_i) and the contiguity/compactness of the individual changed areas.

An empirical space comparison was performed between ESTDM and both uncompressed and compressed forms of the grid snapshot model. Both forms of the snapshot model were implemented through use of the GRASS software package to produce the temporal sequence of gridded images (Shapiro *et al.* 1992). In the uncompressed grid format of GRASS, two bytes of storage are used for each pixel. The compressed grid form utilizes a form of raster run-length encoding, with two bytes used for each new pixel value and two bytes for the corresponding number of contiguous columns (i.e., number of successive x locations) across the scan line containing that same value. Since the total number of pixels in a single scan line never varies, the row value (i.e., y) is not recorded. Thus, the compressed grid format consists simply of a succession of pixel value/x-run-length pairs. Values for x, y, and the component descriptor (i.e., pixel value) in the ESTDM format were each stored in two bytes.

For the sake of comparability between the explicit event structure of ESTDM and the uncompressed and compressed forms of the snapshot model as described above, data representing temporally non-continuous phenomena were used, where an initial base map within the ESTDM-formatted data is not needed. This initial base map is always equivalent to the first snapshot in a snapshot representation of the same data. If there is a small number of events, the differences in space requirements between the compacted snapshot representation and ESTDM would be minimized. As the number of events increases, however, the effect of this single complete image on the total volume of ESTDM-formatted data decreases.

The data used was the historical fire observational data for an area within the Klamath National Forest in extreme North-Central California from 1924 to 1987. These data were supplied by the U.S. Forest Service and consist of forest fire events with an annual temporal resolution and a spatial resolution of 30 m per pixel. Some years had several areas that burned, while in other years, there were no noted forest fire events. Only years where at least one fire event occurred were noted. For each area burned in a given year, a specific fire class value was recorded. The total area covered measured 729 pixels by 832 pixels, or 606 528 pixels in all.

Table 1 shows the comparative sizes for these same data utilizing the compressed and the non-compressed grid snapshot representations and the ESTDM representation for each of the seven fire event years and the total for each over all seven years. It can be seen in table 1 that the compressed snapshot representation for the entire time period required 69 267 bytes in contrast to 8 491 392 bytes required for the uncompressed snapshot representation. The compressed snapshot representation thus provides over a 99 per cent space saving over the uncompressed form. The total storage required for

Table 1. Space comparison between the snapshot representation and ESTDM. The small amount of space used for differing types of header information in the various formats was excluded.

	Uncompressed snapshots	Compressed snapshots	ESTDM
1924	1 213 056	10 341	1948
1926	1 213 056	9 625	706
1929	1 213 056	9 209	160
1954	1 213 056	9 217	184
1955	1 213 056	9 605	676
1966	1 213 056	10 181	1540
1987	1 213 056	11 099	3322
Total	8 491 392	69 267	8536

the same data utilizing the ESTDM is 8 536 bytes, thus requiring only 12·3 per cent of the total space used by the compressed snapshot representation.

5. Potential ESTDM variations

A potential variation on the model just described is the representation of changes in polygon boundaries or other features instead of locations in this temporally-ordered scheme. The design of such a model is not seen as a straightforward task, however. This is primarily due to the problem of how to maintain the integrity of spatial topology as it changes. The basic issue is how to relate changes in *spatial* topology which are not also *temporally* contiguous while using a temporally-based ordering. The solution will require a more complex definition of components within individual events and is an issue for future research.

Another variation on the model just described is to store non-spatial changes as they relate to geographical features. For example, we want to describe a rainfall occurrence within a given geographical region on 12 July 1992 as an event. Suppose in Centre County, the total rainfall associated with the event was 2 inches, and in both Clinton and Huntington Counties the rainfall was 3 inches. Thus the event list would contain a time-stamp of 12 July 1992, and two components: one component containing a component descriptor of 2 inches and a single token consisting of simply the *name* of the feature (i.e., Centre County) or some other unique identifier instead of an array of run-length-encoded locations. Similarly, the second component would contain a component descriptor of 3 inches and two tokens, one for Clinton County and one for Huntington County.

In many potential applications, however, the non-spatial changes relating to geographic features may be more complex, such as would be the case with the reconstruction of a road. Many attributes of the road may have changed such as its width, subsurface characteristics, surface material, traffic capacity, etc. In such a situation, the components associated with individual events would be the road name or other unique identifier. Instead of tokens being associated with this component descriptor, the component descriptor would be used to reference relations within a temporal DBMS. The topic of Temporal DBMS, however, is also currently a very active area of research with much debate ongoing concerning temporal data models (Jensen and Snodgrass 1993, Tansel *et al.* 1993). One particular problem in utilizing the Relational Database Management Systems approach for time-based queries (with or without a spatial dimension) is that there is, by definition of the Relational model, no explicit ordering between relations. Although this has not been mentioned as a problem in initial test implementations of vector-based spatio-temporal data models which employ existing Relational DBMS, such as the one by Hazelton *et al.* (1990), it imposes a significant disadvantage for retrieval and analysis of information on the basis of their temporal order. This problem has been acknowledged among researchers of temporal DBMS (Tansel *et al.* 1993).

6. Summary and future directions

In this paper, a new type of spatio-temporal data model for geographical information systems, called the Event-based SpatioTemporal Data Model (ESTDM) has been described. Unlike existing approaches used in GIS, the ESTDM is designed to explicitly represent change over space relative to time. From a user perspective, a temporally-based representation is needed in order to effectively allow empirical analysis of space/time dynamics, and ultimately allow temporal modelling and simulation of geographical processes as an integrated GIS capability. Because ESTDM explicitly stores change relative to time, procedures for answering queries relating to temporal relationships, as well as analytical tasks for comparing different sequences of change are greatly facilitated.

The use of a time-based representation, such as the one presented here, within a full, GIS-based spatio-temporal analysis context introduces many difficult issues that are beyond the scope of this paper. Much remains to be done before a true temporal GIS can be realized. All of these problem areas are also being addressed by researchers in Temporal DBMS, since that technology is also at an early stage. It thus becomes quickly apparent that there would be much to be gained from interdisciplinary research on temporal databases in general. A review of Temporal DBMS research and how specific areas could be applied in a GIS context has already been given by Langran (1989 a). Two issues directly parallel classical problems in GIS and cartography within the spatial domain. These are temporal generalization and temporal resolution. Temporal resolution is known as the 'granularity' of the database (Jensen *et al.* 1993) within Temporal DBMS, and many of the problems associated with resolution and discretizing values for continuous phenomena as they have become known in the spatial realm are being discovered anew by the Temporal DBMS research community.

Two new issues that are introduced with the addition of the temporal dimension is, first, the distinction between when a state or condition is current or valid in the real world and when that state of condition was entered into the database (i.e., 'valid time' 'transaction time'). These are also known as 'world time' and 'database time', respectively (Jensen *et al.* 1993). This distinction can be important for many applications, but how to effectively represent both types of information within the same database is an open question. The data model described in the current paper therefore was limited to 'world time'. A second major issue unique to temporal databases is how to perform retrospective updates (i.e., inserting new information concerning past conditions or events). This issue involves how to determine which other information already in the database is affected by the new information and must therefore also be changed in order to maintain the integrity of the data. Pioneering research on this issue within the Temporal DBMS community involves the use of a variety of Artificial Intelligence techniques and formal logics for temporal reasoning (Kowalski 1992; Maiocchi *et al.* 1992).

Acknowledgments

This research is sponsored by the National Science Foundation, under Grant FAW NSF 90-27 by the National Aeronautics and Space Administration under Grant NAGW-2686 and by the Forest Service, U.S. Department of Agriculture, under Cooperative Agreement 59-PSW-92-001G. The patient assistance of Elizabeth Wentz in the empirical space comparison is also gratefully acknowledged.

Appendix

The following code in the C programming language shows the actual implementation of ESTDM. Please note that GRASS, a public domain GIS supported by the U.S. Army Corps of Engineers, was used for storing the full-time Base Map. The name of the map type in GRASS is therefore stored within the ESTDM header. The name of the EST series is also used as the manset name in GRASS.

```
/** This is the header file for the ESTDM data structure **/
typedef struct cell_struct{
        short val;                  /* the category value of the component */
        short no_rows;              /* the total number of rows of value val */
        short rows;                 /* the row number of the cell */
        short col1;                 /* the starting column number */
        short col2;                 /* the ending column number */
} aComponent;

typedef struct date_struct{         /* this fits the numerical date into a single word.
                                    It's most commonly used as a single int value */
        short year;
        unsigned char mon;
        unsigned char day;
} aDate;

typedef struct event_struct{
        aDate date                  /* the date associated with the event */
        aComponent *comp;           /* the component associated with the event */
        struct event_struct *next;  /* pointer to the next event struct */
        struct event_struct *prev;  /* pointer to the prev event struct */
} anEvent;

typedef struct header_struct{
        char  map_type[40];         /* the name of the map type/grass map layer */
        char  EST_name[40];         /* the name of the EST series */
        aDate begin;                /* the date of the basemap */
        aDate start;                /* the starting date */
        aDate end;                  /* the ending date */
        anEvent *e_first;           /* pointer to first event in the event list */
        anEvent *e_last;            /* pointer to last event in the event list */
        int   no_evts;              /* the total number of events */
} aHeader;

typedef struct estdm_struct{
        aHeader *hdr;               /*the pointer to current header */
        anEvent *evt;               /* the pointer to current events */
        char**evtnamen-list;        /* the event name list */
} Estdm;
```

References

BURROUGH, P. A., 1986, *Principles of Geographical Information Systems for Land Resources Assessment* (Oxford: Clarendon).

FUJIMOTO, R. M., 1990, Parallel Discrete Event Simulation. *Communications of the ACM*, **33**, 30–53.

GOODCHILD, M. F., GUOQING, S., and SHIREN, Y., 1992, Development and test of an error model for categorical data. *International Journal of Geographical Information Systems*, **6**, 87–104.

GUPTILL, S., 1990, *An Enhanced Digital Line Graph Design* (Circular No. 1048). U.S. Geological Survey.

HAZELTON, N. W. J., 1991, Integrating Time, Dynamic Modelling and Geographical Information Systems: Development of Four-Dimensional GIS, unpublished Ph.D. Dissertation. Department of Surveying and Land Information, The University of Melbourne.

HAZELTON, N. W. J., LEAHY, F. J., and WILLIAMSON, I. P., 1990, On the design of temporally-referenced, 3-D geographical information systems: development of four-dimensional GIS. In *Proceedings of GIS/LIS '90*, Vol. **1**. (ASPRS/ACSM/URISA/AAG: Bethesda MD), pp. 357–372.

HOUGHTON, J. T., JENKINS, G. J., and EPHRAUMS, J. J. eds, 1990, *Climate Change. The IPCC Scientific Assessment* (Cambridge: Cambridge University Press).

JENSEN, C. S., CLIFFORD, J., GADIA, S. K., SEGEV, A., and SNODGRASS, R. T., 1993, A glossary of temporal database concepts. In *Proceedings of the International Workshop on an Infrastructure for Temporal Databases*, (Arlington, TX: Association for Computing Machinery) pp. A25–A29.

JENSEN, C. S., and SNODGRASS, R., 1993, Three proposals for a third-generation temporal data model. In *Proceedings, International Workshop on an Infrastructure for Temporal Databases* (Arlington, TX: Association for Computing Machinery), pp. T1–T10.

KELMELIS, J., 1991, Time and Space in Geographic Information: Toward a Four-Dimensional Spatio-Temporal Data Model. Ph.D. dissertation. The Pennsylvania State University.

KNUTH, D., 1973, *The Art of Computer Programming, Volume III: Sorting and Searching.* (Reading, Mass.: Addison-Wesley).

KOWALSKI, R., 1992, Database updates in the event calculus. *Journal of Logic Programming*, **12**, 121–146.

LANGRAN, G., 1989 a, A review of temporal database research and its use in GIS applications. *International Journal of Geographical Information Systems*, **3**, 215–232.

LANGRAN, G., 1989 b, Time in Geographic Information Systems. Ph.D. dissertation, University of Washington.

LANGRAN, G., 1990, Temporal GIS design tradeoffs. *Journal of URISA*, **2**, 16–25.

LANGRAN, G., 1992, *Time in Geographic Information Systems.* (London: Taylor & Francis).

LANGRAN, G., and CHRISMAN, N. R., 1988, A framework for temporal geographic information. *Cartographica*, **25**, 1–00.

MAIOCCHI, R., PERNICI, B., and BARBIC, F., 1992, Automatic deduction of temporal information. *ACM Transactions of Database Systems*, **17**, 647–688.

PEUQUET, D. J., 1978, Raster data handling in geographic information systems. In *Harvard Papers on Geographic Information Systems*, edited by J. Dutton (Reading, MA: Addison-Wesley), pp. 68–76.

PEUQUET, D. J., 1984, A conceptual framework and comparison of spatial data models. *Cartographica*, **21**, 66–113.

PEUQUET, D. J., 1988, Representations of geographic space: toward a conceptual synthesis. *Annals of the Association of American Geographers*, **78**, 375–394.

PEUQUET, D. J., 1994, It's About Time: A Conceptual Framework for the Representation of Spatiotemporal Dynamics in Geographic Information Systems, *Annals of the Association of American Geographers*, **84**, 441–461.

SHAPIRO, M., WERTERVELT, J., GERDES, D., LARSON, M., and BROWNFIELD, K., 1992, *Grass 4.0 Programmer's Manual*, U.S. Army Construction Engineering Research Laboratory.

TANSEL, A. U., CLIFFORD, J., GADIA, S., JAJODIA, S., SEGEV, A., and SNADGRASS, R., 1993, *Temporal Databases* (Redwood City, CA: Benjamin/Cummings Publishing Co.).