Cubic Map Algebra Functions for Spatio-Temporal Analysis

Jeremy Mennis, Roland Viger, and C. Dana Tomlin

ABSTRACT: We propose an extension of map algebra to three dimensions for spatio-temporal data handling. This approach yields a new class of map algebra functions that we call "cube functions." Whereas conventional map algebra functions operate on data layers representing two-dimensional space, cube functions operate on data cubes representing two-dimensional space over a third-dimensional period of time. We describe the prototype implementation of a spatio-temporal data structure and selected cube function versions of conventional local, focal, and zonal map algebra functions. The utility of cube functions is demonstrated through a case study analyzing the spatio-temporal variability of remotely sensed, southeastern U.S. vegetation character over various land covers and during different El Niño/Southern Oscillation (ENSO) phases. Like conventional map algebra, the application of cube functions may demand significant data preprocessing when integrating diverse data sets, and are subject to limitations related to data storage and algorithm performance. Solutions to these issues include extending data compression and computing strategies for calculations on very large data volumes to spatio-temporal data handling.

Introduction

The term "map algebra" was first introduced in the late 1970s (Tomlin and Berry 1979) and has since been used in loose reference to a set of conventions, capabilities, and analytical techniques that have been widely adopted for raster-based geographic information systems (GIS) (Tomlin 1990; 1991). Map algebra attempts to accommodate a wide variety of GIS applications in a clear and consistent manner by decomposing data, data processing capabilities, and data processing control techniques into elemental components that can then be recomposed with both ease and flexibility. The resulting algebra-like language is one in which single-factor map layers are treated as variables that can be transformed or combined into new variables by way of primitive operations invoked through expressions conforming to a well defined syntax. Map algebra has been incorporated in many GIS and remote sensing image processing packages, and it has been extended in areas ranging from cellular automata (Takeyama and Couclelis 1997) to environmental modeling (van Deursen 1995; Hofierka and Neteler 2001; Pullar 2001) to topographic analysis (Caldwell 2000). It is widely recognized as one of the most influential analytical frameworks for GIS-based raster data handling (Longley et al. 2001; DeMers 2003).

Like most of the analytical frameworks embodied in current GIS packages, map algebra is primarily oriented toward data that are static. Each layer is associated with a particular moment or period of time, and analytical capabilities are intended to deal with spatial relationships. In its original form, map algebra was never intended to handle spatial data with a temporal component. However, as the availability of spatio-temporal data has increased dramatically in recent years due to the growth of satellite remote sensing and other technologies, and as the sophistication of things such as video games and animation in the motion picture industry has raised popular expectations for spatio-temporal processing capabilities there has also been an increasing demand for the spatio-temporal extension of GIS.

One of the reasons for the widespread adoption of conventional map algebra for raster processing in GIS is its simple syntax and the ability to string together multiple functions to create more complex models. These features provide a simple yet powerful toolbox for raster data manipulation analysis. The inclusion of a library of temporal map algebra functions in GIS packages would be just as useful and facile for analyzing the multitude of spatio-temporal raster

Jeremy Mennis, Department of Geography and Urban Studies, 1115 W. Berks Street, 309 Gladfelter Hall, Temple University, Philadelphia, PA 19122. Phone: (215) 204-4748. Email: <jmennis@temple.edu>. **Roland Viger**, Department of Geography, University of Colorado and U.S. Geological Survey, MS 412 Box 25046 DFC, Denver, CO, 80225, USA. Phone: (303) 236-5030 ; Fax: (303) 236-5034. Email: <rviger@usgs.gov>. **C. Dana Tomlin**, Department of Landscape Architecture and Regional Planning, University of Pennsylvania, Philadelphia, PA, 19104-6311. Phone: (978) 724-3352 ; Fax: (215) 573-3770. Email: <tomlin.dana@verizon.net>.

data now being generated. We note that despite the growing volume of research on spatio-temporal data models over the past dozen or so years (e.g., Langran 1992; Peuquet 2001), the extension of map algebra to the temporal dimension has been largely ignored by the spatio-temporal GIS research community. The present research is intended as a first step toward the development of such a temporal map algebra library by providing a conceptual foundation for temporal map algebra and the implementation of a select set of map algebra functions that may be applied to spatio-temporal data.

In the following section a framework for the extension of map algebra to the temporal dimension is described. This design is then demonstrated through a prototype implementation of certain temporal map algebra functions which we call "cube functions." Whereas conventional map algebra functions operate on data layers representing two-dimensional space, cube functions operate on data cubes representing two-dimensional space over a third-dimensional period of time. A case study is used to demonstrate how cube functions can be utilized. This case study analyzes the spatio-temporal variability of remotely sensed, southeastern U.S. vegetation character over various land covers and during different El Niño/ Southern Oscillation (ENSO) phases.

Motivation

A variety of approaches for storing and analyzing spatio-temporal data in GIS have been proposed and implemented (cf. Abraham and Roddick 1999; Peuquet 2001). Most of these approaches have focused on the representation and analysis of spatial phenomena that have a temporal beginning and end, and which may change in their spatial extent, location, or non-spatial properties over their lifetime (Worboys 1994; Wachowicz 1999; Hornsby and Egenhofer 2000). Additionally, conventional metric and topological spatial operators have been extended for spatio-temporal analysis (Breunig 2003). This research has been heavily oriented toward spatial phenomena that are modeled as discrete objects, such as countries or cars, rather than qualities of space that are modeled as continuous fields, such as temperature or population density. Such approaches have generally involved temporal extensions to structured query language (SQL) processing of vector-encoded data (Erwig et al. 1999; Griffiths et al. 2001).

In contrast there have been fewer efforts at developing representation and analysis strategies for spatio-temporal field-like data stored in raster format. The most straightforward approach for storing spatio-temporal raster data is the "snapshot" model in which change through time is represented using a series of spatially registered grids, each corresponding to a particular moment in time (Langran 1992). Another approach employs a form of spatio-temporal "run length encoding" in which temporal events that mark changes in the value of individual grid cells are stored (Peuquet and Duan 1995). Other approaches have used object-oriented techniques to store spatio-temporal arrays of environmental measurements encoded in a spatio-temporal "data cube" in which two cube dimensions are spatial and the third is temporal (Raper and Livingstone 1995; Mennis 2003).

The statistical analysis of spatio-temporal raster data has been undertaken for interpolation (Mitasova et al. 1995), geostatistics (Kyriakidis and Journel 1999; Christakos 2000), and Bayesian approaches (Christakos et al. 2002). Researchers in remote sensing have also applied spectral analysis and principle components analysis to extract temporal signals from time series of imagery (Eastman and Fulk 1993; Jakabauskus et al. 2001). Other researchers focusing on spatio-temporal raster processing have addressed feature extraction, for instance in the recognition of meteorological phenomena from time series of observational meteorological data, remotely sensed imagery, or general circulation model (GCM) output (Stolorz et al. 1995; Yuan 2001; Mennis and Peuquet 2003). Van Deursen (1995) and Wesseling et al. (1996) describe a language for extending the spatial capabilities of map algebra to support dynamic modeling, implemented in the GIS PCRaster.

Also of relevance to the present research is three-dimensional raster GIS in which space is exhaustively partitioned into a regular tessellation of three-dimensional cubic volume elements (voxels) instead of two-dimensional square grid cells (Raper 1989; 2002). This approach has been used for a variety of geologic and atmospheric science modeling and visualization applications (Hibbard et al. 1994; Marschallinger 1996; Masumoto et al. 2004). Of note is the Geographic Resources Analysis Support System (GRASS) GIS, originally developed by the U.S. Army Construction Engineering Research Laboratories, which supports three-dimensional raster encoding for which a limited set of map algebra functions are available (Brown et al. 1997; Neteler 2004). Other GIS software packages that handle raster data, such as ArcGIS (Environmental Systems Research Institute, Inc.), provide the ability to link multiple grids in a "stack" to mimic a three-dimensional representation, though the analysis capabilities associated with these stacks are typically limited. Raper (2002) provides a taxonomy of three-dimensional spatial query and analysis functions.

Research in image processing has also described algorithms similar in nature to those used in three-dimensional GIS. These algorithms have focused on the extension of filters, image arithmetic, and segmentation to three and four dimensions (Nikolaidis and Pitas 2000). This research has been applied primarily to medical imaging (Wan and Higgins 2003) as well as to motion detection in video sequencing (Rajagopalan et al. 1997) and time series of satellite imagery (Yamomoto et al. 2001).

Despite this breadth of research in spatio-temporal and three-dimensional GIS there is a notable lack of attention given to extending map algebra beyond two spatial dimensions, and particularly to the temporal dimension. In fact, none of the research cited above explicitly addresses the extension of map algebra algorithms to the temporal dimension. Because of the lack of general-purpose, spatio-temporal raster data manipulation capabilities in current commercial GIS, researchers working with spatio-temporal raster data have been forced to develop customized algorithms (e.g., Yuan 2001; Mennis and Peuquet 2003). Consequently, these algorithms are specific to the application domain under investigation and cannot be readily reused in other application contexts.

In the present research we demonstrate how the original map algebra construct can be extended to support spatio-temporal raster data analysis in a manner that transcends any one particular application domain. For this purpose we draw from previous research in spatio-temporal GIS in the use of the spatio-temporal cube metaphor for data encoding and manipulation. The cube function algorithms are adapted from conventional map algebra and its extension to three spatial dimensions, as well as from related functions associated with three-dimensional image processing.

Extending Map Algebra for Spatio-Temporal Processing

Two-Dimensional Map Algebra

In the original map algebra, each variable or "layer" is a bounded plane surface on which position is expressed in terms of Cartesian (X,Y) coordinates. The portion of that surface uniquely referenced by a given X,Y pair is termed a "location," and each location is associated with (what is usually just) one recorded characteristic such as soil type, land value, or population. Each of the distinct conditions depicted on a layer is termed a "zone," and each zone is represented by way of a numerical value that may relate to a nominal, ordinal, interval, ratio, or cyclical scale of measurement. Every map algebra operation accepts one or more of these variables as input and generates a single new variable as output. Just as conventional algebra operations such as addition and subtraction can be combined to form complex equations, map algebra operations can also be combined by repeatedly using the output from one as input to another. Thus, while individual operators are narrowly defined, the range of possibilities for combining these operations is entirely open ended.

Several dozen primitive operations have been defined, and these are typically organized into three major groups referred to as "local," "focal," and "zonal" functions (DeMers 2002). Local functions compute a new value for every location as a function of that location's value(s) on one or more existing layers. Focal functions compute each location's new value as a function of the existing values, distances, and/or directions of neighboring locations on an existing layer. These focal functions can be applied to neighborhoods that are defined in terms of travel costs or lines of sight as well as physical separation. Note that we include "incremental" functions—which were originally assigned to their own separate group—within focal functions as they are algorithmically very similar. Zonal functions compute each location's new value as a function of the values from one existing layer (the value layer) which are associated with that location's zone on another existing layer (the zone layer).

Three-Dimensional Map Algebra

To extend map algebra into a three-dimensional world (i.e., one in which each X,Y location is also associated with a full range of Z positions) is conceptually straightforward. Locations that had been associated with squares are now associated with cubes, and this results in cubic forms of zones, neighborhoods, and layers. Figure 1 demonstrates how each of the conventional local, zonal, and focal map algebra functions can be transformed into three-dimensional form.

The three-dimensional extension of local functions presents few difficulties except, perhaps, for the need to envision the combination of multiple cubes as a process by which those cubes are "blended into" one another rather than "superimposed" (Figure 1a). In the case of the focal functions, the addition of a third spatial dimension means that neighborhoods are now a bit more interesting (and challenging computationally) in terms of their geometry, but the existing functions are quite amenable to this extension (Figure 1b). For example, whereas the original FocalProximity function would have generated a series of concentric rings of distance around specified X,Y locations, it now generates concentric *shells* of distance around locations in X,Y,Z space. Or, consider a focal function that computes the slope at each location given a grid of elevations. In the three dimensional case, that surface is replaced by what might be envisioned as a smoky room in which smoke intensity varies from one location to another. The analogous three-dimensional focal function wouldn't measure slope, per se, but rather the rate at which smoke intensity (as opposed to elevation) varies between neighboring locations. Such focal functions are similar in principle to filter operations in three-dimensional image processing (Nikolaidis and Pitas 2000). For three-dimensional zonal functions, zones might be thought of more in terms of polyhedra than polygons. Instead of operating on a value layer and a zone layer, a three-dimensional zonal function would operate on a value cube and a zone cube (Figure 1c).

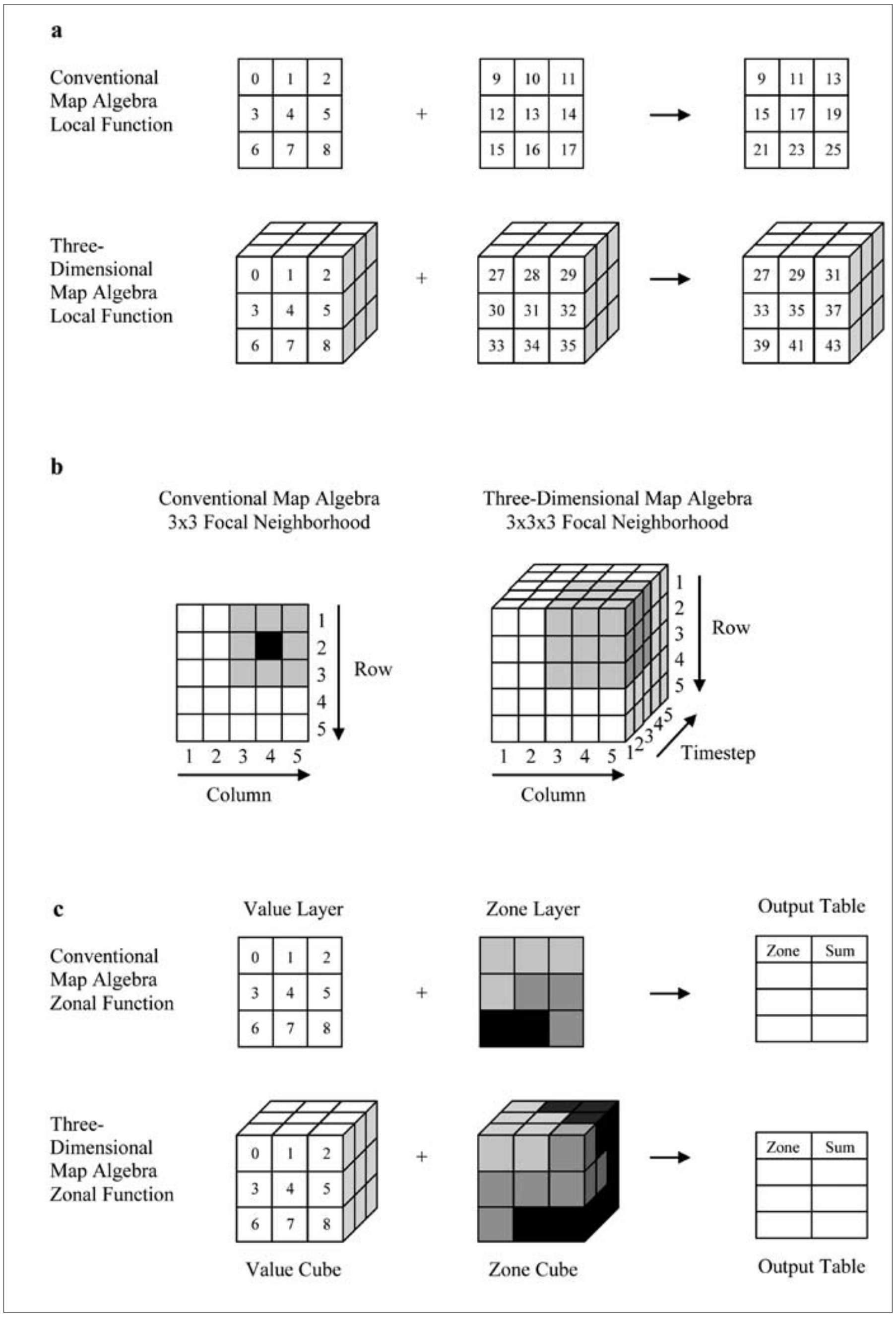


Figure 1. Comparisons between conventional and three-dimensional local (a), focal (b), and zonal (c) map algebra functions. In (b), the conventional map algebra drawing shows a 3x3 focal neighborhood centered on the [row column] position (2,4), which is colored black. The analogous three-dimensional map algebra drawing shows a 3x3x3 focal neighborhood centered on the [row,column,timestep] position (2,4,2), which is obscured by outer cube elements.

Map Algebra in the Temporal Dimension

Given these map algebra functions on X,Y,Z space, a number of temporal cartographic modeling capabilities can be realized by simply regarding the Z dimension as temporal in nature. On the other hand there are important differences between a third spatial dimension (Z) and the temporal dimension (T). Whereas the Z dimension often embodies a directional bias associated with gravity, the T dimension *always* embodies a directional bias as, inevitably, time marches on. Since the units of T will also differ from those of X and Y, these too must now be independently specified.

For local functions, the introduction of a T dimension raises the prospect of combining values by spatio-temporal correspondence, spatial correspondence only, or temporal correspondence only. In the first case, the combination of two X,Y,T cubes would be identical to the combination of two X,Y,Z cubes as shown in Figure 1a: the values of one cube would be combined with those of the corresponding values in the other cube, and the output would be an X,Y,T cube of similar dimensions. In the second case, an X,Y,T cube could be combined with an X,Y layer by matching each X,Y,T cube element with its X,Y layer counterpart based solely on the X,Y coordinate values of each element. Likewise, an X,Y,T cube could be combined with a time series data set having no spatial dimension, by matching each X,Y,T cube element with its time series counterpart based solely on the T coordinate value of each element.

Local functions on X,Y,T cubes may also support the ability to summarize a subset of the dimensions by another. For example, given a single X,Y,T input cube, one may wish to combine all the values in the T dimension associated with each individual location, and the output would be an X,Y layer. Analogously, one may wish to combine all the values in the X and Y dimensions associated with each individual moment in time, and the output would be a time series of values.

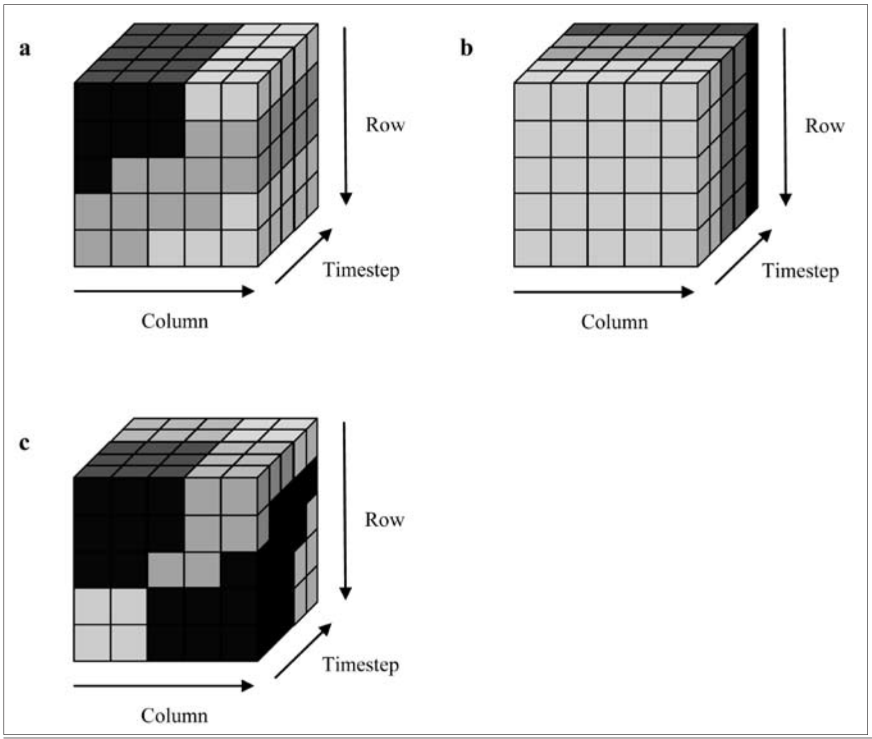


Figure 2. Types of zone cubes used in spatio-temporal zonal functions: a zone cube that varies over space but not time (a), a zone cube that varies over time but not space (b), and a zone cube that varies over both space and time (c).

Perhaps the most fundamental distinction between an X,Y,Z and an X,Y,T environment from the perspective of map algebra, however, is the need to define both zones and neighborhoods not only in terms of space-time units but also in terms of units that relate solely to the spatial or temporal dimension. These neighborhoods could also be anisotropic in space and time, so that, say, the value of a spatio-temporal X,Y,T position is calculated as a function of the values that have previously occurred at that location. Consider, for example, a form of FocalMaximum in which the value of an X,Y,T position is calculated as the maximum precipitation value that has occurred in the last year.

For zonal functions, spatio-temporal data also present several new opportunities. If we assume the value cube in a zonal function is spatio-temporal, the zone cube may be spatial (varying in space but not time) (Figure 2a), temporal (varying in time but not space) (Figure 2b), or spatio-temporal (varying over both space and time) (Figure 2c). If the zone cube is spatial, the output would report a summary of the spatio-temporal values in the value cube for each spatial zone. If the zone cube is temporal, the output would report a summary of the spatio-temporal values in the value cube for each temporal zone. Similarly, for a spatio-temporal zone cube, the output would report a summary for each spatio-temporal zone.

Implementation of Cube Functions

As a first step toward the implementation of the temporal map algebra described above, we have implemented a number of new functions that operate on a three-dimensional data model in which two dimensions represent planimetric position and the third dimension represents time. These are called "cube functions" and are referred to by adding the prefix "cube" to the conventional function’s name, e.g., FocalMean becomes cubeFocalMean. We will describe the cube function versions of LocalSum, FocalSum, and ZonalSum as examples. These were implemented using the Interactive Data Language (IDL) (Research Systems, Inc.), which was selected as a rapid prototyping tool because of its high-level array-handling capabilities.

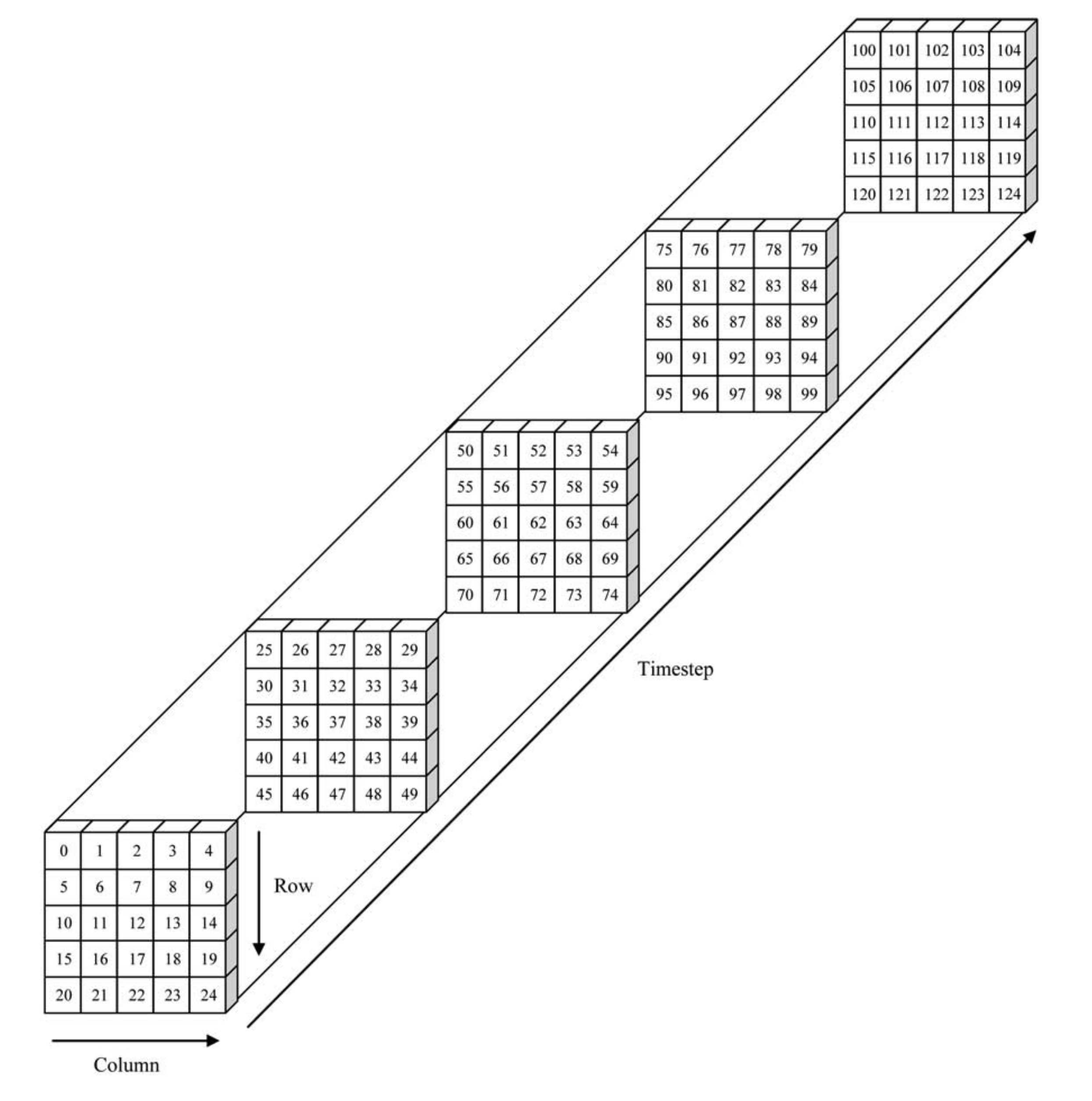
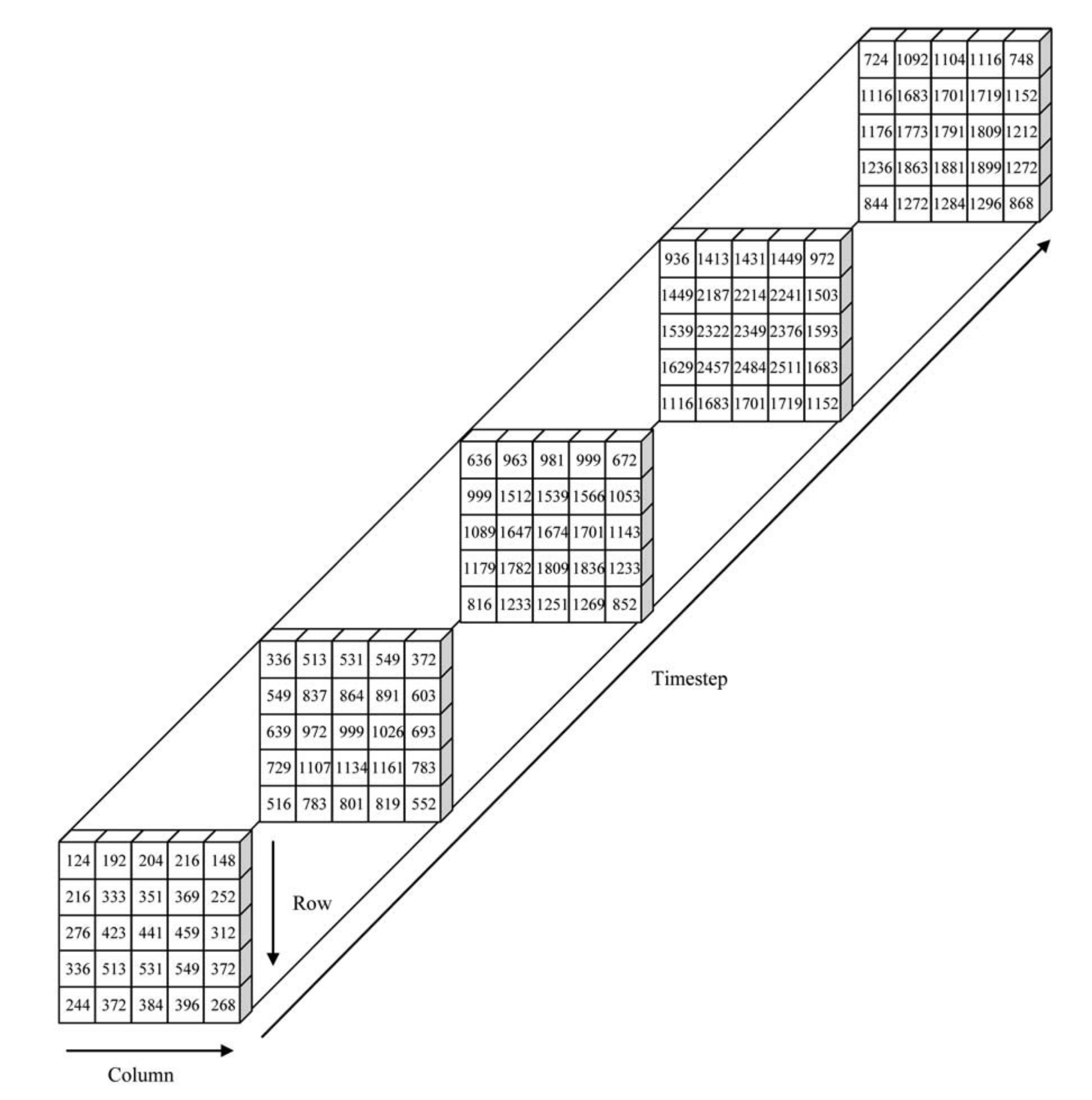


Figure 3. A [5,5,5] data cube.

The spatio-temporal data structure we developed is an implementation of the space-time cube concept as described above. For the purpose of prototyping the cube functions, a spatio-temporal "data cube" was encoded as a simple three-dimensional array of the form [*row*, *column*, *timestep*] where the row and column array positions imply location in the spatial dimensions and the timestep array position implies "location" in the temporal dimension. One thematic value was encoded for each unique position in the three-dimensional array. An individual position in the three-dimensional array is referred to as an "element" of the data cube.

Local cube functions were implemented by using the basic mathematical operators (e.g., +, -, /, *) of IDL, which supports their application to multi-dimensional arrays. Focal and zonal cube functions were implemented by extending the looping programming structures that iterate over two-dimensional arrays in conventional map algebra to three dimensions. We illustrate focal and zonal cube functions using synthetic spatio-temporal data sets which are small enough to be validated graphically. Focal functions are demonstrated using the [5,5,5] data cube shown in Figure 3. These data could be air temperatures sampled over time or any field-like geographic phe-



124	192	204	216	148
216	333	351	369	252
276	423	441	459	312
336	513	531	549	372
244	372	384	396	268

336	513	531	549	372
549	837	864	891	603
639	972	999	1026	693
729	1107	1134	1161	783
516	783	801	819	552

636	963	981	999	672
999	1512	1539	1566	1053
1089	1647	1674	1701	1143
1179	1782	1809	1836	1233
816	1233	1251	1269	852

936	1413	1431	1449	972
1449	2187	2214	2241	1503
1539	2322	2349	2376	1593
1629	2457	2484	2511	1683
1116	1683	1701	1719	1152

724	1092	1104	1116	748
1116	1683	1701	1719	1152
1176	1773	1791	1809	1212
1236	1863	1881	1899	1272
844	1272	1284	1296	868

Figure 4. Results of the cubeFocalSum function, when applied to the value cube shown in Figure 3.

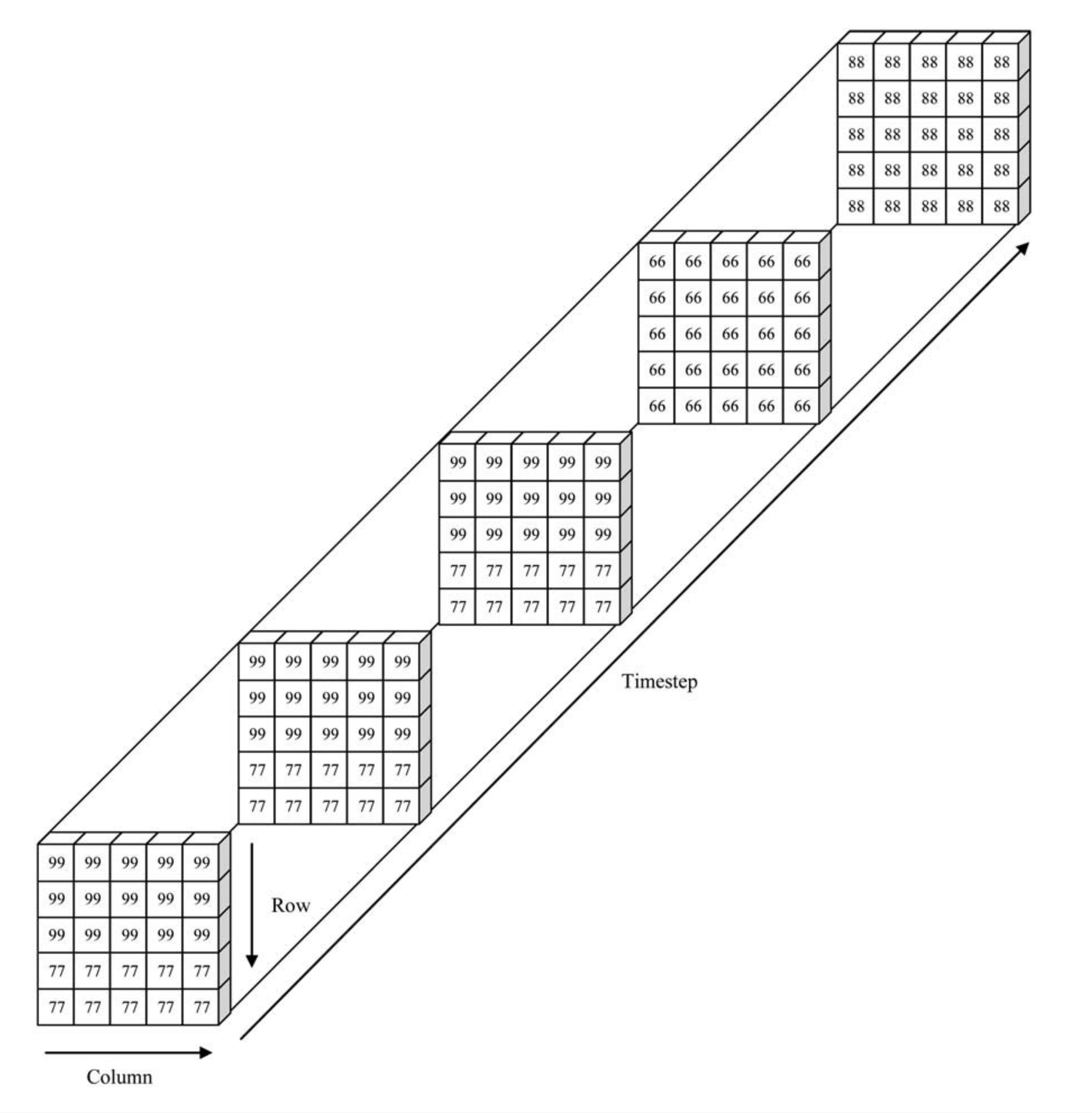
nomena regularly sampled over two-dimensional space and time. Figure 4 reports the results produced by applying a cubeFocalSum function using a [3,3,3] element focal neighborhood, i.e., the focal neighborhood extends over a three-row-by-three-column-by-three-timestep "cube" that is centered over the focal kernel.

Although not demonstrated in this example, cube focal neighborhoods can also be delineated in purely spatial or purely temporal dimensions. Thus, a focal function can be applied to its spatial surroundings, its temporal context, or its vicinity in both time and space. To illustrate the zonal cube functions, a cubeZonalSum function was applied using the zone data cube shown in Figure 5 and the value data cube shown in Figure 3. Table 1 reports the results where each row in the table shows the sum of the values of all elements in the value cube that lie within each of the spatio-temporal zones defined in the zone cube.

Case Study: Analysis of Spatio-Temporal Remote Sensing Data

Overview

To demonstrate the utility of temporal map algebra, we conducted an analysis of 1982-1992 El



88	88	88	88	88
88	88	88	88	88
88	88	88	88	88
88	88	88	88	88
88	88	88	88	88

66	66	66	66	66
66	66	66	66	66
66	66	66	66	66
66	66	66	66	66
66	66	66	66	66

99	99	99	99	99
99	99	99	99	99
99	99	99	99	99
77	77	77	77	77
77	77	77	77	77

99	99	99	99	99
99	99	99	99	99
99	99	99	99	99
77	77	77	77	77
77	77	77	77	77

99	99	99	99	99
99	99	99	99	99
99	99	99	99	99
77	77	77	77	77
77	77	77	77	77

Figure 5. A [5,5,5] zone cube. Each element's value encodes membership in a particular zone.

Niño/Southern Oscillation (ENSO)-related vegetation change over different land covers in the southeast USA, including North Carolina, South Carolina, and Georgia. Cube functions were used to manipulate, integrate, and analyze data that are spatial (land cover data), temporal (ENSO phase data), and spatio-temporal (vegetation change data). Vegetation "greenness," or intensity, was indicated using Normalized Difference Vegetation Index (NDVI) data derived from the Advanced Very High Resolution Radiometer (AVHRR), a satellite-borne earth sensor. These eight-km-resolution data were transformed prior to the analysis by first calculating the monthly mean for each pixel and then subtracting that monthly mean from each pixel's observed NDVI value (Figure 6), a common approach taken in climatological analyses. Vegetated land cover data for the study regions were acquired from the U.S. Geological Survey's Geographic Information Retrieval and Analysis System (GIRAS) program (Figure 7). These data record land cover using the two-stage hierarchical Anderson classification system (Anderson et al. 1976), denoted here as Land Cover 1 and Land Cover 2 (Table 2).

Data concerning ENSO were acquired from the National Oceanic and Atmospheric Administration

Zone	Sum
66	2175
77	1335
88	2800
99	1440

Table 1. The results of the cubeZonalSum function applied to the value cube shown in Figure 3, summarized by the zone cube shown in Figure 5.

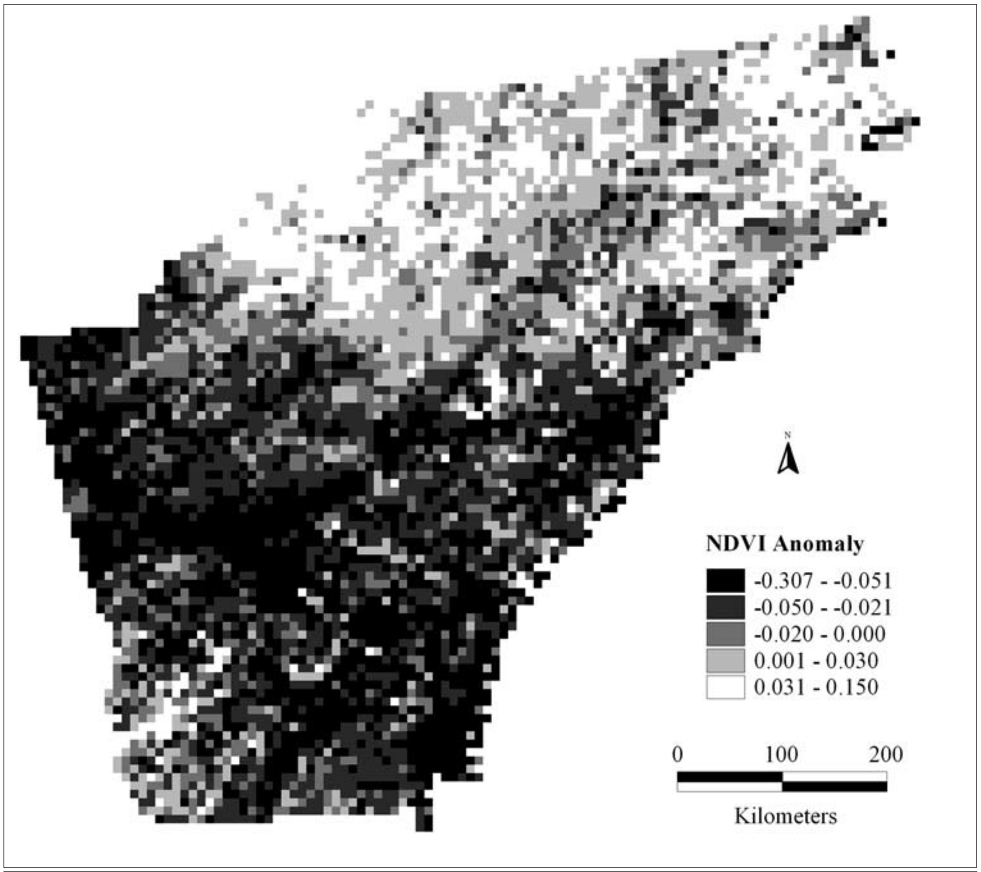


Figure 6. Southeast USA NDVI anomaly for May 1983, an ENSO warm phase month with associated strongly negative NDVI anomalies, particularly in the southern part of the study region.

(NOAA) (Smith and Sardeshmukh 2000). These data indicate whether each month of the study period is classified as being associated with an ENSO warm phase (El Niño), cold phase (La Niña), or neutral phase. This classification was derived using standardized anomalies of both the Southern Oscillation Index (SOI) and AVHRR-derived Sea Surface Temperature (SST) data from Nino 3.4, a region of the equatorial Pacific Ocean that indicates ENSO phase. Two ENSO-phase classifications were used, ENSO 1 (Table 3) and ENSO 2; the former incorporates a slightly more restrictive definition of a warm or cold phase event.

Analysis

The NDVI anomaly, ENSO 1, ENSO 2, Land Cover 1, and Land Cover 2 data were used to populate six individual data cubes: **`NDVI`**, **`ENSO1`**, **`ENSO2`**, **`LC1`**, and **`LC2`**, respectively. For the `NDVI` cube, each element encodes the NDVI anomaly from the monthly climatologic mean. These values are unitless and range from approximately -0.50 to 0.50. For the `ENSO1` and `ENSO2` cubes, each element encodes a -1, 0, or 1, indicating whether that month is associated with an ENSO cold phase, neutral phase, or warm phase, respectively. Elements in the `LC1` and `LC2` cubes encode the numeric IDs of the Land Cover 1 and Land Cover 2 land cover classes, respectively (e.g., "4" for forest land and "41" for deciduous forest).

Two cubeZonalMean functions were then applied using NDVI as the value data cube and ENSO1 and ENSO2 as the zone data cubes, respectively, in order to investigate whether different ENSO phases have different mean NDVI anomaly values. This analysis demonstrates the adaptation of the conventional map algebra zonal function for spatio-temporal analysis, as described above, where zones vary in time but not space (Figure 2b). Such a function therefore produces a table where each row summarizes the NDVI anomaly data for a different temporal zone.

Results are presented in Table 4. For both ENSO phase classifications, an ENSO warm phase is associated with a negative NDVI anomaly and an ENSO cold phase is associated with a positive NDVI anomaly. Of the two ENSO phase classifications, ENSO 1 captures the greater magnitude in the NDVI anomaly associated with ENSO warm and cold phase events.

In order to investigate the response of vegetation to ENSO over different land covers, each of the ENSO phase classification data cubes (ENSO1 and ENSO2) is combined with each of the land cover data cubes (LC1 and LC2) to create four new data cubes: **ENSO1_LC1**, **ENSO2_LC1**, **ENSO1_LC2**, and **ENSO2_LC2**. The ENSO phase classification and land-cover data cubes were combined using a local function that multiplied the ENSO data cube's value (i.e., -1, 0, or 1) by 100 and then added the numeric land-cover code ID to the result to produce a unique numeric identifier for each land cover/ENSO phase combination.

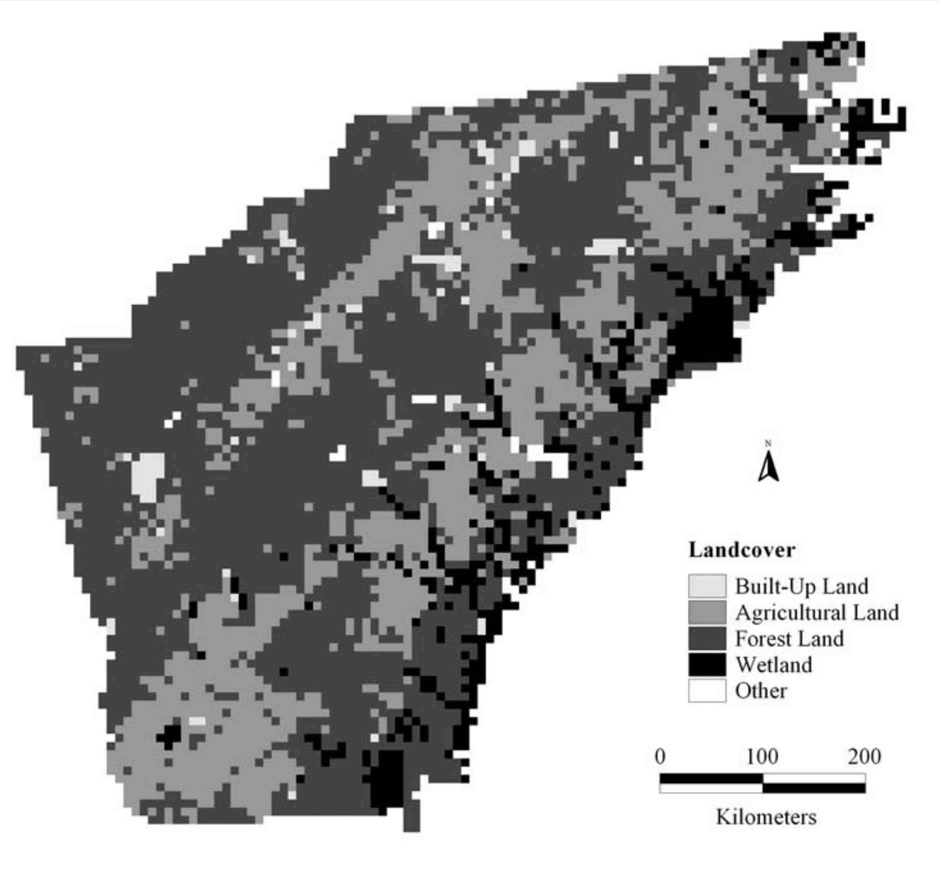


Figure 7. Southeast USA land cover. Only level 1 of the Anderson land cover classification is shown for simplicity.

ID	Land Cover 1	Count	ID	Land Cover 2	Count
2	Agricultural Land	1,698	21	Cropland and Pasture	1,696*
4	Forest Land	3,274	41	Deciduous Forest	796
			42	Evergreen Forest	1,079
			43	Mixed Forest	1,399
6	Wetland	440	61	Forested Wetland	371
			62	Non-Forested Wetland	69

Note: ID is the numeric code for each land cover. Count indicates the number of grid cells with that land cover. * Two out of 1,698 agricultural land grid cells are assigned to a different Land Cover 2 class and are not considered in the analysis.

Table 2. Vegetated land cover classes analyzed in the case study.

For example, consider a cube element position with an ENSO1 value of 1 (ENSO warm phase) and a LC1 value of 4 (forest land). In the cube resulting from the local function used here that element would have a value of 104 (1*100+4). The value of 100 was used because none of the land-cover code values exceed 100, thus ensuring a unique value for each land cover–ENSO phase combination. This adaptation of the conventional map algebra local function was discussed initially above and is shown diagrammatically in Figure 1a, where two data cubes are combined. Note that whereas the ENSO phase data only vary temporally (such as that depicted as in figure 2B), and the land cover data only vary spatially (such as that depicted in Figure 2a), the new combined data cubes contain zones that extend over both space and time (such as that depicted as in Figure 2c).

A series of cubeZonalMean functions were then applied using NDVI as the value cube and each of the four combined ENSO phase/land cover data cubes as the zone cubes. As discussed above, because these zone cubes are spatio-temporal, the result of each of these functions is a table in which each row summarizes the NDVI anomaly data for each spatio-temporal zone.

Table 5 reports the results of these cubeZonalMean functions where the ENSO1_LC1 and ENSO2_LC1 cubes were used as the zone cubes. All of the individual land covers maintained the same general

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1982	0	0	0	0	0	0	1	1	1	1	1	1
1983	1	1	1	1	1	0	0	0	0	0	0	0
1984	0	0	0	0	0	0	0	0	0	0	0	0
1985	0	0	0	0	0	0	0	0	0	0	0	0
1986	0	0	0	0	0	0	0	0	0	0	0	0
1987	0	0	0	1	1	1	1	1	1	1	1	0
1988	0	0	0	0	0	0	0	0	-1	-1	-1	0
1989	0	0	0	0	0	0	0	0	0	0	0	0
1990	0	0	0	0	0	0	0	0	0	0	0	0
1991	0	0	0	0	0	1	1	1	1	1	1	1
1992	1	1	1	1	1	1	0	0	0	0	0	0

Note: ENSO warm phase = ‘1’, ENSO cold phase = ‘-1’, and ENSO neutral phase = ‘0’.

Table 3. ENSO 1 Classification of months into ENSO phase.

ENSO phase-NDVI anomaly pattern as exhibited in Table 4 where the ENSO warm (cold) phase exhibits a negative (positive) NDVI anomaly. Wetlands exhibit the greatest ENSO warm and cold phase anomalies from the long-term mean, followed by forest land and then agricultural land. Table 6 presents the results of the cubeZonalMean functions where ENSO1_LC2 and ENSO2_LC2 cubes were used as the zone cubes. The results show that of the two wetlands sub-classes, it is the non-forested wetlands that are contributing the most to the negative vegetation response during ENSO warm phase events. Non-forested wetlands also have by far the highest positive anomaly associated with ENSO cold-phase events.

ENSO Classification	Warm Phase	Neutral Phase	Cold Phase
ENSO 1	-0.024	0.008	0.034
ENSO 2	-.0019	0.010	0.024

Table 4. Results of the cubeZonalMean function applied to the NDVI data cube and ENSO1 and ENSO2 data cubes.

In addition to the mean of the NDVI anomaly within each ENSO phase/land cover class, it is also instructive to consider variance as an indication of consistency in vegetation response to ENSO; a small variance indicates a consistent response while a large variance indicates a less consistent response. To investigate this, we applied a series of cubeFocalVariance functions on the NDVI anomaly data cube. Recall that the user has the ability to set the spatial and temporal parameters for the three-dimensional X,Y,T neighborhood used in cube focal functions. We experimented with four different neighborhood configurations: a [3,3,1] spatial-only neighborhood; a [1,1,3] temporal-only neighborhood; a [3,3,3] spatio-temporal neighborhood; and a [11,11,11] spatio-temporal neighborhood. Note that while the use of a [1,1,3] neighborhood calculates the variance of just three values (the element upon which the function is centered in addition to the elements that immediately precede and follow it), such a neighborhood provides information on the nature of the temporal variation in a NDVI anomaly value over a three month period. A series of cubeZonalMean functions was then used to summarize the resulting NDVI anomaly variance data cubes using the ENSO1_LC2 data cube.

Results are reported in Table 7. The wetlands sub-classes had the highest mean NDVI anomaly variance by far, with non-forested wetlands having a much greater value than forested wetlands. Generally, the ENSO neutral phase had the greatest mean

	Land Cover 1	Warm Phase	Neutral Phase	Cold Phase
ENSO 1:	Agricultural Land	-0.019	0.007	0.034
	Forest Land	-0.025	0.009	0.033
	Wetland	-0.029	0.010	0.039
ENSO 2:	Agricultural Land	-0.016	0.009	0.020
	Forest Land	-0.020	0.010	0.025
	Wetland	-0.023	0.011	0.035

Table 5. Results of the cubeZonalMean function applied to the NDVI data cube and the ENSO1_LC1 and ENSO2_LC1 data cubes.

	Land Cover 2	Warm Phase	Neutral Phase	Cold Phase
ENSO 1:	Cropland and Pasture	-0.019	0.007	0.034
	Deciduous Forest	-0.026	0.009	0.026
	Evergreen Forest	-0.027	0.009	0.041
	Mixed Forest	-0.024	0.008	0.031
	Forested Wetland	-0.026	0.009	0.031
	Non-Forested Wetland	-0.043	0.014	0.082
ENSO 2:	Cropland and Pasture	-0.016	0.009	0.020
	Deciduous Forest	-0.020	0.010	0.022
	Evergreen Forest	-0.021	0.011	0.026
	Mixed Forest	-0.019	0.010	0.025
	Forested Wetland	-0.020	0.011	0.025
	Non-Forested Wetland	-0.034	0.010	0.089

Table 6. Results of the cubeZonalMean function applied to the NDVI data cube and the ENSO1_LC2 and ENSO2_LC2 data cubes.

NDVI anomaly variance, with the noted exceptions of the two wetlands sub-classes, in which the cold phase produced the highest values. Results derived from using the [1,1,3] neighborhood to generate the NDVI anomaly variance data cube were similar to those using the [3,3,1] neighborhood, though mean NDVI anomaly variance values of the former were generally higher. This suggests that NDVI anomaly tends to vary more from one month to the next in a given location than from one location to an adjacent neighbor at a given point in time. As expected, mean NDVI anomaly variance was lower for the NDVI anomaly variance data cube generated using the [3,3,3] spatio-temporal neighborhood than for that generated using the [11,11,11] spatio-temporal neighborhood.

Discussion

Although the focus of this paper is the extension of map algebra for spatio-temporal analysis, and not ENSO-related vegetation dynamics, it is enlightening to offer a brief interpretation of the results of the case study to demonstrate the utility of the techniques described here. This analysis agrees with previous research (Mennis 2001) in finding that in the southeast USA the colder and wetter climate associated with ENSO warm phases tends to suppress vegetation greenness, while the warmer and dryer conditions associated with ENSO cold phases enhance vegetation greenness. This pattern is expected, given the association of ENSO warm phase events with lower temperatures and greater precipitation than is typical in the southeast USA (Ropelewski and Halpert 1986; Mote 1996). The amplified ENSO signal observed for non-forested wetlands may be due to the flooding of these areas as a result of increased precipitation associated with an ENSO warm phase. This tends to suppress NDVI values. In addition, biomass in non-forested wetlands is concentrated in more transitory vegetation that is more prone to disruption by weather events than, say, woodlands. This interpretation is supported by the analysis of the NDVI anomaly variance, which demonstrates high spatial and temporal variability in the vegetation character of non-forested wetlands.

Note that the case study analysis, while certainly possible without the use of cube functions, would otherwise demand significant manual operations and preprocessing that would be both cumbersome and time consuming. Consider, for instance, how the case study may have been undertaken using conventional GIS techniques. For this task, we assume the spatial data are in a standard raster format, as they were prior to being imported to the spatio-temporal data structure we developed. The NDVI anomaly data were initially stored as a set of individual spatial data layers, each layer representing a particular month and year. The Land Cover 1 and Land Cover 2 data were stored as two individual spatial data layers, and the ENSO 1 and ENSO 2 data were stored as two text files.

Table 4 could thus be derived without using cube functions by extracting those NDVI anomaly data layers corresponding to the different ENSO phases, exporting and aggregating those data from the various layers into a single table for each ENSO phase, and then calculating the mean NDVI anomaly for each ENSO phase. Tables 5 and 6 could be generated without using cube functions by overlaying each NDVI anomaly data layer with the land-cover data layer, encoding the ENSO phase for each NDVI anomaly observation (though all observations within each layer would have the same ENSO phase value), and generating a unique identifier for each ENSO phase/land cover classification. The data from each NDVI anomaly data layer would then need to be either exported and appended, or somehow combined in a single file, so that the mean of each unique ENSO phase and land cover class combination could be calculated.

Generating Table 7 without cube functions would require perhaps the most intensive GIS user input as compared to the cube function approach. In order to complete the cubeFocalVariance function using a spatio-temporal neighborhood, the set of NDVI anomaly observations within a neighborhood would have to be extracted from the appropriate NDVI anomaly data layers and the variance calculated. Such an operation would have to be performed for every iteration of the focal function, and it would be particularly cumbersome for neighborhoods with large temporal extents. Once the variance for the neighborhood of each NDVI anomaly observation has been calculated and used to populate a new set of spatial data layers, the procedure specified above could be used to summarize the mean variance for each unique ENSO phase and land-cover class combination.

There are perhaps other approaches to performing the case study analysis without cube functions, such as by combining the time series of NDVI anomaly values into a single data layer. However, all conventional GIS approaches to the case study analysis would require extensive processing in which the GIS analyst must perform a series of data and file manipulation operations. With cube functions, however, the entire case study analysis may be completed using just a few simple cube function statements. Of course, it takes some effort to import the data into the spatio-temporal data structure to which the cube functions are applied. We developed a file reader script that reads in spatio-temporal data formatted in a text file, where each row in the file represents a single-element spatial location and each column represents a moment in time. An inline header encodes the number of rows and columns, as well as a value for indicating if a particular element has a "no data" value. Generating this input file demanded some preprocessing in a commercial GIS package, though far less than the preprocessing necessary to perform the analysis without the use of cube functions.

We note that one could write scripts to automate some of the manual file manipulations and other data processing required were the analysis done using a commercial GIS package. In essence, the idea of cube functions is that they are such scripts already written and made available to the GIS user. However, it is important to note that were a GIS user to script a solution to the case study analysis presented above, the scripts would likely be specific to the task at hand, and not generalizable to other application domains or data sets. Cube functions, on the other hand, are intended to be generic and applicable to a wide variety of data that can be encoded within a three-dimensional data structure such as the one we have implemented here. And like conventional map algebra, temporal map algebra functions can be accessed via simple statements that can be combined to produce more sophisticated modeling and analysis for a variety of application domains.

Conclusion

We recognize that the cube data structure and functions implemented in this research are algorithmically relatively straightforward. Their purpose, however, is to provide an algorithmic and syntactic framework for the development of a rich library of temporal map algebra functions. Given this framework, many of the other conventional map algebra functions can be extended to their cube function analogs. For example, though the current implementation of the cube functions supports only a handful of basic statistical operators such as sum and mean, more sophisticated operators can be incorporated without any major revision of the basic code. Likewise, though the current implementation of the focal function only supports a "cubic" spatio-temporal neighborhood (with user-defined spatial and temporal extent), a variety of neighborhood options, such as spatio-temporal cones or "doughnuts," can easily be implemented and incorporated into the structure of the algorithm.

Like the conventional map algebra, our approach to temporal map algebra and prototype implementation of the cube functions engenders certain assumptions and limitations. We discuss four of these issues in this paper. First, just as conventional map algebra is intended to work with two-dimensional raster grids and not vector data, cube functions are intended to handle three dimensional raster data cubes. While the raster data model can be used to represent nearly any kind of geographic phenomena, it is typically used for field-like geographic phenomena (Goodchild 1992). While it is certainly possible to represent moving objects, such as cars, in the spatio-temporal data cube, the cube functions are not intended to analyze this type of phenomena, except in situations where such phenomena may be easily represented as individual zones in a cube zonal function.

Second, just as conventional map algebra assumes uniform intervals between grid cells in a raster, cube functions assume uniform spatial and temporal intervals between elements in the spatio-temporal data cube. In addition, cube functions assume that the spatial and temporal extent and resolution of all the data sets that are entered into a given function are identical. Meeting these assumptions would likely demand significant data preprocessing for many "real-world" applications, in which the integration of data from diverse sources, and hence various extents and resolutions, is the norm. In most GIS packages which support map algebra it is up to the user to ensure the algorithms are being applied appropriately; and our approach to the implementation of cube function places the same onus on users. However, we acknowledge that the introduction of a temporal component complicates users' ability to meet these assumptions. For instance, spatial interpolation and re-sampling algorithms available in a variety of GIS packages may be used to transform multiple two-dimensional raster data sets to a single resolution for entry into conventional map algebra functions. However, the spatio-temporal analogs of these algorithms are not widely available.

Third, the utility of cube functions is dependent on the ability of users to transform their spatio-temporal data into the data structure on which the cube functions operate. The approach we used in the prototype implementation was to develop a file reader script for a particular format which we found to be of use because of our own experiences and software expertise. However, spatio-temporal data may be stored in a variety of formats. Therefore, it is essential to ultimately develop a file reader that can ingest many different file formats to populate the spatio-temporal data structure.

The fourth limitation of the cube function approach implemented here concerns related issues of data storage efficiency and algorithm performance. Raster encoding of continuous data is typically costly in terms of data storage because of the need to store an individual value for each location in the raster. The problem is exacerbated in spatio-temporal raster data where data sets of even moderate spatial extent can place significant data storage demands. One approach to this problem is to adapt conventional raster compression methods, such as run-length encoding and quadtrees (Samet 1990; Worboys 1995), to three dimensions (Samet 1990; Peuquet and Duan 1995). Data volume also has an impact on algorithm performance, as cube function algorithms are often required to iterate over all elements in a data cube and are thus computationally intensive for very large data sets. This is particularly true for cube focal functions, which can require data retrieval in the spatial and/or temporal dimensions at each iteration of the algorithm.

As a related problem, the current prototype implementation of certain cube functions require that multiple blocks of memory, each equal to the size of the input data cube, be assigned simultaneously. This can severely limit the size of data sets that can be handled by the cube functions, particularly when using a desktop computer, as we were for this research. This latter performance issue may potentially be addressed using "out-of-core" computing in which computation is applied to data residing on disk as opposed to in memory (Kandemir et al. 1997), and which has been applied to handling very large spatial data sets (Ferreira et al. 2000; Pitas and Cotsaces 2000). This approach should work well for cube function algorithms in which algorithmic calculations may be easily compartmentalized by space and/or time.

ACKNOWLEDGEMENTS

The authors wish to thank Michael Worboys and Katherine Hornsby for coordinating the NSF workshop that spurred this research. Thanks also to Donna Peuquet and others who provided comments on previous drafts of the manuscript. This research was supported by NASA NIP grant #NAG5-12598.

REFERENCES

Abraham, T., and J.F. Roddick. 1999. Survey of spatio-temporal databases. *Geoinformatica* 3: 61-9.

Anderson, J.R., E.E. Hardy, J.T. Roach, and R.E. Witmer. 1976. *A land use and land cover classification system for use with remote sensor data*. U.S. Geological Survey Professional Paper No. 964.

Breunig, M. 2003. *On the way to component-based 3D/4D geoinformation systems*. Berlin: Springer.

Brown, W.M., H. Mitasova, and L. Mitas. 1997. Design, development and enhancement of dynamic multidimensional tools in a fully integrated fashion with the GRASS GIS. Technical report, US-Army CERL.

Caldwell, D. 2000. Extending map algebra with flag operators. In: *Proceedings of the Fifth International Conference on GeoComputation*. WWW and CD.

Christakos, G. 2000. *Modern spatiotemporal geostatistics*. New York, New York: Oxford University Press.

Christakos, G., P. Bogaert, and M. Serre. 2002. *Temporal GIS: Advanced functions for field-based applications*. Berlin: Springer.

DeMers, M.N. 2002. *GIS modeling in raster*. Chichester, U.K.: John Wiley and Sons.

DeMers, M.N. 2003. *Fundmentals of Geographic Information Systems, Second Edition*. Chichester: John Wiley and Sons.

Eastman, J. R., and M. Fulk. 1993. Long sequence time series evaluation using standardized principal components. *Photogrammetric Engineering and Remote Sensing* 59: 1307-12.

Erwig, M., R.H. Guting, M. Schneider, and M. Vazirgiannis. 1999. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. *Geoinformatica* 3: 269-96.

Ferreira, R., G. Agrawal, R. Jin, and J. Saltz. 2000. Compiling data intensive applications with spatial coordinates. In: *Proceedings of the 13th International Workshop on Languages and Compilers for Parallel Computing*, pp. 339-54.

Goodchild, M.F. 1992. Geographical data modeling. *Computers and Geosciences* 18: 401-8.

Griffiths, T., A.A.A. Fernandez, N.W. Paton, K.T. Mason, B. Huang, M. Worboys, C. Johnson, and J. Stell. 2001. Tripod: A comprehensive system for the management of spatial and aspatial historical objects. In: *Proceedings of the 20th International Conference on Conceptual Modeling*, pp. 84-102.

Hibbard, W., B. Paul, D. Santek, C. Dyer, A. Battaiola, and M-F. Voidrot-Martinez. 1994. Interactive visualization of Earth and space science computations. *Computer* 27: 65-72.

Hofierka, J., and M. Neteler. 2001. Volume modeling of soils using GRASS GIS 3D tools. In: *Proceedings of the Second Italian GRASS User's Meeting*, Trento, Italy, February 1-2.

Hornsby, K., and M.J. Egenhofer. 2000. Identity-based change: A foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science* 14: 207-24.

Jakubauskas, M.E., D.R. Legates, and J. Kastens. 2001. Harmonic analysis of time-series AVHRR NDVI data. *Photogrammetric Engineering and Remote Sensing* 67: 461-70.

Kandemir, M., J. Ramanujam, and A. Choudhary. 1997. Improving the performance of out-of-core computations. In: *Proceedings of the International Conference on Parallel Processing*.

Kyriakidis, P.C., and A.G. Journel. 1999. Geostatistical space-time models. *Mathematical Geology* 31: 651-84.

Langran, G. 1992. *Time in geographic information systems*. London, U.K.: Taylor and Francis.

Longley, P.A., M.F. Goodchild, D.J. Maguire, and D.W. Rhind. *Geographic Information Systems and Science*. Chichester: John Wiley and Sons.

Marschallinger, R. 1996. A voxel visualization and analysis system based on AutoCAD. *Computers and Geosciences* 22: 379-86

Masumoto S., V. Raghavan, G. Yonezawa, T. Nemoto, K. Shiono. 2004. Construction and visualization of a three dimensional geologic model using GRASS GIS. *Transactions in GIS* 8: 211-23.

Mennis, J. 2001. Exploring relationships between ENSO and vegetation vigour in the south-east USA using AVHRR data. *International Journal of Remote Sensing* 22: 3077-92.

Mennis, J. 2003. Derivation and implementation of a semantic GIS data model informed by principles of cognition. *Computers, Environment, and Urban Systems* 27: 455-79.

Mennis, J., and D.J. Peuquet. 2003. The role of knowledge representation in geographic knowledge discovery: A case study. *Transactions in GIS* 7: 371-91.

Mitasova, H., L. Mitas, B.M. Brown, D.P. Gerdes, I. Kosinovsky. 1995. Modeling spatially and temporally distributed phenomena: New methods and tools for GRASS GIS. *International Journal of Geographic Information Science* 9: 443-6.

Mote, T.L. 1996. Influence of ENSO on maximum, minimum, and mean temperatures in the southeast United States. *Physical Geography* 17: 497-512.

Neteler, M. (ed.). 2004. *GRASS 5.0 programmer's manual*. Trento, Italy: ITC.

Nikolaidis, N., and I. Pitas. 2000. *3D image processing algorithms*. Hoboken, New Jersey: John Wiley and Sons.

Peuquet, D.J. 2001. Making space for time: Issues in space-time data representation. *Geoinformatica* 5: 11-32.

Peuquet, D.J., and N. Duan. 1995. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems* 9: 7-24.

Pitas, I., and C.I. Cotsaces. 2000. Memory efficient propagation-based watershed and influence-zone algorithms for large images. *IEEE Transactions on Image Processing* 9: 1185-99.

Pullar, D. 2001. MapScript: A map algebra programming language incorporating neighborhood analysis. *GeoInformatica* 5: 145-63.

Rajagopalan, R., M.T. Orchard, and R.D. Brandt. 1997. Motion field modeling for video sequences. *IEEE Transactions in Image Processing* 6: 1503-16.

Raper, J.F. (ed.). 1989. *Three dimensional representations in geographical information systems*. London, U.K.: Taylor and Francis.

Raper, J. 2002. *Multidimensional geographic information science*. New York, New York: Taylor and Francis.

Raper, J.F., and D. Livingstone. 1995. Development of a geomorphological data model using object-oriented design. *International Journal of Geographical Information Systems* 9: 359-83.

Ropelewski, C.F., and M.S. Halpert. 1986. North American precipitation and temperature patterns associated with the El Niño/Southern Oscillation (ENSO). *Monthly Weather Review* 114: 2352-62.

Samet, H. 1990. *Design and analysis of spatial data structures*. Reading, Mass.: Addison Wesley.

Smith, C.A., and P. Sardeshmukh. 2000. The effect of ENSO on the intra-seasonal variance of surface temperature in winter. *International Journal of Climatology* 20: 1543-57.

Stolorz, P., H. Nakamura, E. Mesrobian, R.R. Muntz, E.C. Shek, J.R. Santos, J. Yi, K. Ng, S.Y. Chien, R. Mechoso, and J.D. Farrara. 1995. Fast spatio-temporal data mining of large geophysical datasets. In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. pp. 300-05.

Takeyama, M., and H. Couclelis. 1997. Map dynamics: Integrating cellular automata and GIS through geoalgebra. *International Journal of Geographical Information Science* 11: 73-91.

Tomlin, C.D. 1990. *Geographic information systems and cartographic modeling*. Englewood Cliffs, New Jersey: Prentice Hall.

Tomlin, C.D. 1991. Cartographic modelling. In: *Geographic information systems: Principles and applications*. Essex, U.K.: Longman Scientific & Technical. pp. 361-74.

Tomlin, C.D., and J.K. Berry. 1979. A mathematical structure for cartographic modeling in environmental analysis. In: *Proceedings of the American Congress on Surveying and Mapping*, pp. 269-83.

van Deursen, W.P.A. 1995. Geographical information systems and dynamic models: Development and application of a prototype spatial modelling language.Unpublished Ph.D. Dissertation, Utrecht University, Netherlands.

Wachowicz, M. 1999. *Object-oriented design for temporal GIS*. London, U.K.: Taylor and Francis.

Wan, S.-Y., and W.E. Higgins. 2003. Symmetric region growing. *IEEE Transactions on Image Processing* 12: 1007-15.

Wesseling, C.G., D. Karssenberg, W.P.A. Van Deursen, and P.A. Burrough. 1996. *Integrating dynamic environmental models in GIS: The development of a dynamic modelling language*. *Transactions in GIS* 1: 40-48.

Worboys, M.F. 1994. A unified model for spatial and temporal information. *The Computer Journal* 37: 27-34.

Worboys, M.R. 1995. *GIS: A computing perspective*. London, U.K.: Taylor and Francis.

Yamomoto, T., H. Hanaizumi, and S. Chino. 2001. A change detection method for remotely sensed multispectral and multitemporal images using 3-D segmentation. *IEEE Transactions on Geosciences and Remote Sensing* 39: 976-85.

Yuan, M. 2001. Representing complex phenomena with both object- and field-like properties. *Cartography and Geographic Information Science* 28: 83-96.

■