ELSEVIER

Landscape and Urban Planning 30 (1994) 3–12

LANDSCAPE AND URBAN PLANNING

Map algebra: one perspective

C. Dana Tomlin

Department of Landscape Architecture and Regional Planning, Graduate School of Fine Arts, The University of Pennsylvania, Philadelphia, PA 19104, USA

Abstract

Map algebra is a general set of conventions, capabilities, and techniques that have been widely adopted for use with geographic information systems (GISs). This paper presents several broad and introductory views of map algebra that attempt to place this approach in conceptual and historical context. These views focus respectively on the map algebraic data model, its data-processing construct, notational syntax, predecessors, and likely near-term future.

Keywords: Data modeling; GIS; Map algebra

1. Introduction

It does have a certain ring to it. Not unlike 'GIS' (geographic information system) itself, 'map algebra' is one of those terms that can roll off the tongue and into accepted parlance with unquestioning ease and unquestioned authority. Also like GIS, however, this too is a term which, on closer inspection, can generate both confusion and controversy. For some, map algebra refers to no more than a style of notational syntax. For others, it refers to particular tools, methods, and even theories of geographic analysis.

This paper refers to map algebra as the term was first introduced in the late 1970s (Tomlin and Berry, 1979) to describe a particular set of conventions, capabilities, and techniques for the analysis of digital cartographic data. To illustrate briefly its nature and use, let us consider a simple, if unlikely, example. In Fig. 1 is a 'cartographic' image that depicts ten black circles and two black squares, all lying on a background of white. Notice that one circle that lies almost directly between the two squares. It is not really a circle; it is more like an egg. And it is not directly between those squares; it is slightly offset to one side. In fact, they are not actually squares at all; they are more like trapezoids. And yet, you do see the 'circle that lies almost directly between the two squares'.

Well, if you can see it, your GIS should certainly be able to distinguish that particular circle as well. Right? In map algebraic terms, this 'simple' task would be addressed as follows:

(1) Transform the original image into perpendicular rows and columns of very small dark and light 'dots'.

(2) Generate a new image on which each conterminous 'island' of dark dots (i.e. one of the squares or circles) is uniquely identified.

(3) Determine how much of each island's perimeter is associated with each dot (assuming that each of those on the edge of an island will be associated with a certain measurable frontage).

SSDI 0169-2046(94)05001-I

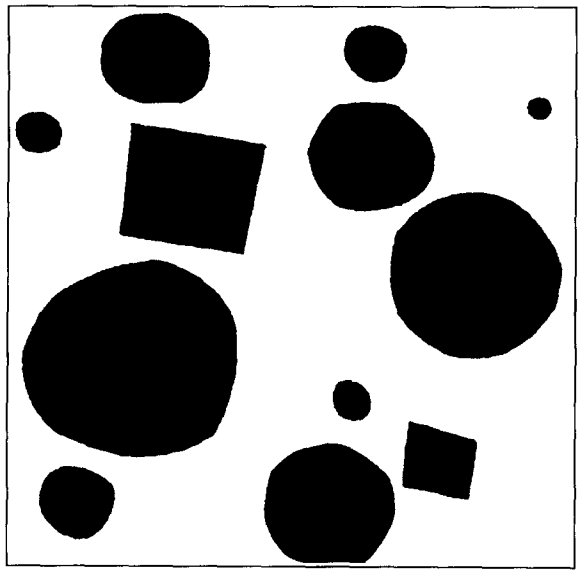

Fig. 1. A typical cartographic image. A 'map' of 12 distinct cartographic features provides the basis for a simple query that serves to illustrate the nature and use of map algebra: 'Which of the circles lies between the two squares?'

(4) Add these increments of perimeter to determine the total for each island.

(5) Compute the square of each island's perimeter.

(6) Determine how much surface area is associated with each dot (assuming that those along the edge of each island will occupy less than those in the center).

(7) Add these increments of surface area to determine the total for each island.

(8) Compute the ratio of each island's squared perimeter to its surface area.

(9) Distinguish 'circles' (islands with ratios of less than 8:1) from 'squares' (islands with ratios of 8:1 or greater).

(10) Generate an image showing just one of the two squares.

(11) Calculate every dot's proximity to that square.

(12) Generate an image showing the second of the two squares.

(13) Calculate every dot's proximity to that second square.

(14) Calculate the sum of every dot's two proximity values.

(15) Generate an image indicating, for each circle, an average of the summed proximity values for all dots within it.

(16) Select the circle with the lowest average.

Although that may seem like a lot of work to do what you were able to do at a glance, note that each of the steps is completely mechanical: the sort of thing (unlike a human glance) that can easily be done by machine. Note, too, that each step performs a general-purpose function: the sort of thing that might well find use in a number of other contexts. And note that this procedure could well be packaged and applied repeatedly to much larger problems: perhaps images of 10 000 or 10 000 000 circles. It is qualities such as these that make map algebra useful in the context of a GIS.

This paper examines map algebra not from the inside out but the outside in. In contrast to fuller descriptions of the language (Tomlin, 1990, 1991), it steps back just far enough for a shift in focus from content to context. That context is then presented as seen from five different points of view. Respectively, these deal with map algebraic variables, operations, expressions, origins, and prospects.

2. Map algebraic variables: what's where or where's what?

One of the major components of any information system is a body of recorded facts, a set of data to which meanings can be ascribed and from which inferences can be drawn. The manner in which such facts are organized can have a subtle and yet profound effect on these meanings and inferences. Consider, for example, the difference between a route map and a step-by-step set of directions or a listing of prices by item as opposed to a listing of items by price.

Geographic information systems are generally designed to deal with facts relating to the surface of the Earth, and most of these systems organize their data in terms of metaphorical maps. Although this may seem only natural in keeping with cartographic tradition, the use of maps in this context is only one of many possibilities. In

digital form, geographic data can just as readily be stored, manipulated, and presented in the form of relational tables, graphs, words, sounds, animations, and so on.

Notwithstanding this range of possibilities, map algebra also employs a map metaphor in organizing data. All data associated with a particular geographic area of interest are stored in the form of what amount to single-factor maps: bounded plane surfaces on which each location is associated with exactly one recorded characteristic. All such 'maps' are taken to represent the same geographic area at the same scale, orientation, and planimetric projection. Termed 'layers', these constitute the only type of map algebraic variable.

So what about numbers, tables, and other common forms of digital data? In map algebra, all exist in the (virtual if not actual) form of one or more map layers. A scalar is a layer on which every location is associated with the same numerical value. A table of attributes associated with soil types, for example, is regarded as a set of layers: one for soil pH, another for soil stability, another for percolation rate, and so on. Although these layers may not all 'exist' to the same degree of reality in actual bits and bytes, all are seen in the mind's eye of the user as single-factor maps.

Each of the characteristics depicted on a layer is termed a 'zone', and each zone is represented by way of a numerical value. This use of numbers rather than (or in addition to) colors and graphic symbols to represent geographic characteristics is one of the major distinctions between map algebraic variables and traditional cartographic maps.

Another important distinction relates to the representation of space. On traditional maps, the cartographic plane is partitioned by way of lines, patterns, symbols, labels, and other graphic devices. Often, these devices rely on considerable interpretation 'by eye'. A point in the plane is part of a wetland because it is near a symbol depicting a tuft of aquatic vegetation. Another point is at an elevation defined by its position relative to the two nearest topographic contour lines. In a digital mapping system intended for data storage and retrieval only, this type of traditional graphic device will often suffice. In a GIS intended for analytical use, however, the cartographic plane must be partitioned in unambiguous terms.

This, in turn, raises questions relating to cartographic precision. On traditional maps, spatial precision is determined by factors such as line width, sheet size, and the practical limits of human vision. In a GIS, on the other hand, precision has little to with the physical dimensions of any particular display. It relates instead to the size and shape of the smallest unit of cartographic space for which data can be recorded. If such units are referenced by way of Cartesian (i.e. equally spaced and perpendicular row–column or x, y) coordinates, as they are in most GISs, then each unit is a discrete square of finite dimensions that are defined by the smallest increments along the respective coordinate axes. In the earlier example of the circle between two squares, such units were referred to as dots. In more general map algebraic terms, each is called a location.

In a raster or image-based GIS, a location would correspond to a grid cell or pixel. This would be true regardless of whether the system were to employ quadtree techniques (partitioning the cartographic plane into quadrants, subquadrants, sub-subquadrants, and so on), run-length methods (grouping adjacent locations of similar value into parallel strips), or any other form of raster-encoding device. In each case, the device merely serves to represent efficiently groups of discrete locations.

In a vector or drawing-based GIS, each location would correspond to the area uniquely associated with any x, y point, and this would also be true regardless of data-encoding devices. In the vector case, these devices include such things as line segments, chains (of line segments), arcs, circles, polygons, and networks. Here again, each device merely serves to represent efficiently groups of locations. Said in more evocative terms, a circle is not a circle but a set of locations. And said in more provocative terms, a circle is a set of locations that are discrete: not infinitely small but only as small as can be addressed by the finite precision of the x, y coordinate system with which that line's endpoints have been defined.

What makes that statement provocative is the issue of spatial precision. When a circle (or a line, a chain, an arc, a polygon, etc.) is decomposed into atomistic units, are not certain geometric properties lost? The answer is both yes and no. Consider, for example, a cartographic object defined as a circle with a 1 m radius. Note that the area and perimeter of this object can already be established with infinite precision, regardless of the spatial coordinate system involved. We have specified form independent of position and have thereby been able to avoid any issue of finite spatial resolution. Suppose, however, that the circle is centered on location (0, 0). Does it encompass location (0, 1.1)? The answer depends on just what is meant by 'location (0, 0)'. Does that mean anywhere within 0.5 m of location (0, 0) in the x and the y dimensions? Or does it mean anywhere within 0.05 m, 0.005 m, 0.000 000 000 000 005 m, or an even finer level of spatial precision? Only when this question of spatial resolution is answered can the question of spatial position be addressed. And once a finite level of resolution is established, should not it be imposed on measures of form as well as position?

The answer to that open question will generally depend on the nature of the spatial phenomena involved. Are those phenomena manifest in spatial 'objects' such as points, lines, polygons, or networks? Or are they manifest in spatial 'qualities' such as distance, direction, narrowness, density, rate of change, or degree of interspersion? If those phenomena can best be expressed as objects by recording 'where' as a function of 'what', then measures of form and position can remain independent. If they are better expressed as 'qualities', however, by recording 'what' as a function of 'where', then form and position must both be defined in terms of discrete spatial units.

Map algebra adopts the latter view, to retain generality. Whereas (the presence or absence of) 'objects' can be expressed as 'qualities', the converse is not often true. By expressing all kinds of punctual, lineal, areal, and surficial phenomena in conceptually simple and similar terms (i.e. numerical values associated with discrete cartographic locations), the map algebraic data model not only makes it possible to interrelate them, but to do so with a small set of tools.

3. Map algebraic operations: the worm's eye view

If the first major component of an information system is its data, the second component is a set of data-processing capabilities. In most geographic information systems, these capabilities are associated with: (1) programming tasks such as getting into and out of particular processing environments, creating or invoking stored procedures, conditional looping, error handling, or control of peripheral devices; (2) data preparation tasks such as line digitizing, video scanning, map editing, image enhancement, cartographic reprojection, or translation of file storage formats; (3) data presentation tasks such as map drawing, charting and graphing, report generation, preparation of tables, or production of live animations; (4) tasks that are associated with data interpretation.

Interpretation in this context refers to the transformation of data into information. Data, in this regard, are viewed as recorded facts of general purpose and potential utility. Information, on the other hand, embodies facts of more specialized purpose and actual utility in some context. It is this ability to interpret, to translate meanings and inferences from implicit to explicit form, that distinguishes an information system such as a GIS from its cousins in fields such as remote sensing, automated mapping, computer-assisted drafting and design, database management, and so on.

Whereas any practical use of map algebra will almost certainly involve capabilities associated with programming, data preparation, and data presentation, the algebra itself relates only to tasks of data interpretation. Here, interpretive capabilities are organized much like data: in elementary yet complementary units. Each of these data-processing units is a map algebraic operation which, like any conventional algebraic operation, accepts one or more variables as input and generates a single variable as output. In this

case, however, the variables involved are not merely numbers but layers. And the operations involved are not merely arithmetic or trigonometric but cartographic in nature.

Given map algebraic variables that represent geographic characteristics with numerical values, it is possible to process those characteristics with mathematical functions. And given that these characteristics are associated with atomistic locations, it is possible to define such functions in terms of individual locations. In other (and more expressive) words, it is possible to adopt a worm's eye point of view. In contrast to the conventional bird's eye perspective, from which conventional maps are normally viewed and most GIS operations defined, this atomistic perspective lends both clarity and generality.

To illustrate this important distinction, consider again the problem of locating that circle between the two squares. What does it mean to be 'more or less circular' or 'more or less square' or 'almost directly in between' two designated areas? From the bird's eye perspective, such spatial relationships are easy enough to describe in English but are sometimes difficult to translate into a language of geometric measurements more meaningful to a computer.

From the worm's eye perspective, on the other hand, things are much simpler. As illustrated in Fig. 2, a location is part of a square (more or less) if it is part of a conterminous group of locations whose squared-perimeter-to-area ratio is greater than 8:1. As illustrated in Figs. 3 and 4, the degree to which a location lies 'in between' two squares can be measured precisely in terms of the sum of its respective distances from those squares. And the degree to which a circle lies 'in between' can be determined by simply averaging these summed distance values for all of that circle's locations.

From the worm's eye vantage, a full complement of cartographic analysis and synthesis capabilities can be expressed in terms of only four types of map algebraic operation. Respectively, these employ what are termed local, zonal, focal, and incremental processing functions.

Local operations compute a new value for every location as a function of the existing value(s) that are associated with that location on one or more specified layers. This function may, for example: count the number of dissimilar values associated with each location; uniquely identify different combinations of existing values; explicitly replace existing values with specified new values; compute arithmetic sums, differences, products, ratios, or roots; calculate trigonometric sines, cosines, tangents, arc sines, arc cosines; report statistical means, medians, maxima, minima, and modes. In a number of geographic information systems, what are referred to as map algebraic capabilities are in fact limited to this first and most basic of the four major classes of map algebraic operation.

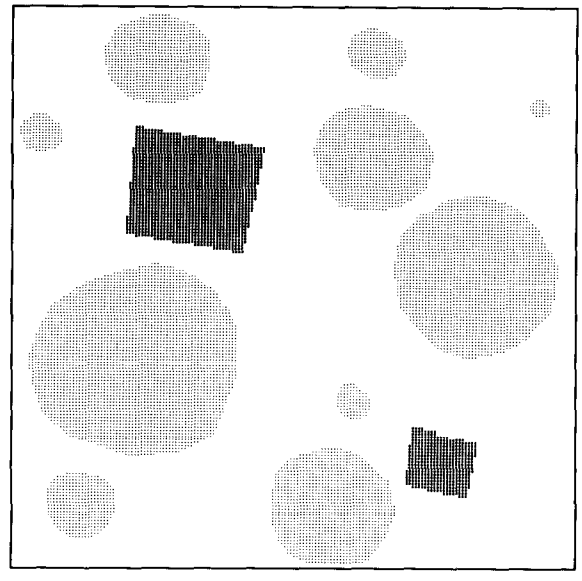

Fig. 2. Circles and squares. To distinguish the 'circles' from the 'squares' shown in Fig. 1, that image is first expressed as a field of discrete row–column locations. The amount of area and perimeter associated with each of these locations is then calculated, and these two figures are respectively tallied for each conterminous 'island'. If the ratio of squared perimeter to area for any island is less than eight, that island is more circular than square.

In the second class are zonal operations, each of which computes a new value for every location as a function of the existing values from one specified layer that are associated with that location's zone on another specified layer. Much like their local counterparts, these functions may,

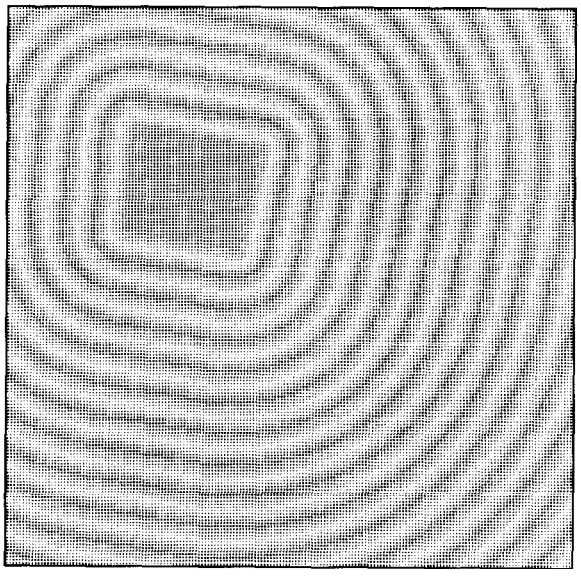

Fig. 3. Proximity to one of the squares. Alternating shades of gray indicate zones of distance around one the two 'squares' depicted in Fig. 2.

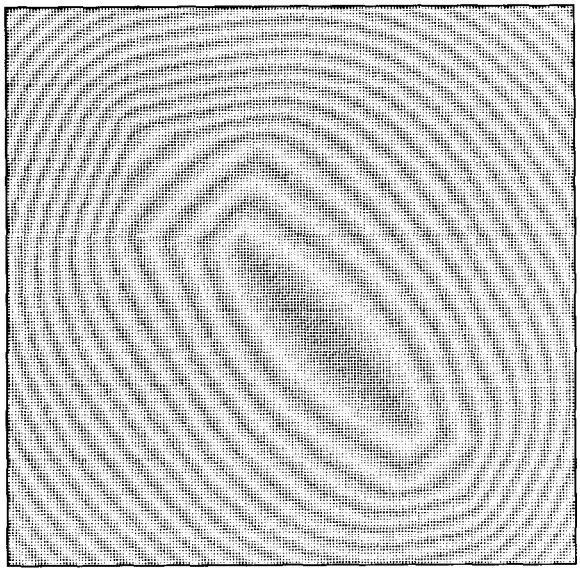

Fig. 4. Between the two squares. Shades of gray indicate the degree to which each location lies between the two 'squares' shown in Fig. 2. This map layer was generated by adding each location's distance from one square (as one shown in Fig. 3) to its distance from the other.

for example: count the number of dissimilar values associated with each zone; identify different combinations of existing values within each zone; explicitly replace combinations of existing values with specified new values; calculate sums, products, means, medians, maxima, minima, and modes; compute the relative magnitude of each location's value as compared with others in its zone. Most geographic information systems are surprisingly weak in this area.

Focal operations are those which compute a new value for every location as a function of the existing values, distances, and/or directions of neighboring locations on a specified layer. Like both local and zonal counterparts, these operations may: employ functions that count the number of dissimilar values associated with each location's neighborhood; uniquely identify different combinations of existing values within each neighborhood; explicitly replace combinations of existing values with specified new values; calculate sums, products, means, medians, maxima, minima, modes, and distance-weighted averages; compute the relative magnitude of each location's value as compared with others within its neighborhood; report the distance, the bearing, or the value of nearest neighbors; uniquely identify conterminous groups of locations in a common zone. Furthermore, these functions may be applied to neighborhoods that are defined in terms of travel costs or lines of sight rather than physical distance. Although virtually all geographic information systems provide some degree of neighborhood or focal processing, only a few have even begun to explore the potential of extending familiar functions of Euclidean distance and/or direction into non-Euclidean space.

Finally, the incremental operations compute a new value for every location as a function of its lineal, areal, or surficial form on a specified layer. These operations may indicate each location's length or shape as part of a lineal network; its surface area, frontage, or shape as part of an areal pattern; or its slope, aspect, drainage direction(s), or volume as part of a surficial form. In most geographic information systems, this type of capability is limited to surface processing.

Just as conventional algebraic operations such

as addition and subtraction can be combined to form complex equations, map algebraic operations can also be combined by repeatedly using the output from one as input to another. Thus, although individual functions are narrowly defined, the range of possibilities for combining functions is open ended.

4. Map algebraic expressions: medium vs. message

Given data and a set of data-processing capabilities, the major remaining component of any information system is a mechanism for processing control. In a geographic information system, this mechanism may involve keystrokes, graphic 'point-and-click' gestures, or even spoken words. It will also probably involve the use of a system-specific language. Map algebra attempts to relate to as many of these languages as possible by employing a simple and highly general form of verbal notation.

Map algebraic expressions, much like their conventional algebraic counterparts, are imperative statements in declarative form that specify sequences of operations and the variables to which they apply. The general syntax for these expressions can be illustrated by example. In Fig. 5 is a map algebraic expression of the procedure that was used to isolate that circle between the two squares.

In some ways, the syntactic characteristics of this notation are entirely cosmetic. What is here given as the 'FocalProximity' operation, for example, is just as effectively referred to in the respective command languages of several different geographic information systems as 'Spread',

EachBlob	= FocalInsularity	of *Blobs*	
BlobEdge	= ZonalSum	of (IncrementalFrontage of *EachBlob*) within *EachBlob*	
BlobSize	= ZonalSum	of (IncrementalArea of *EachBlob*) within *EachBlob*	
BlobShape	= LocalRatio	of (LocalProduct and *BlobSize*	of *BlobPerimeter* and *BlobPerimeter*)
Squares	= LocalRating	of *BlobShape*	with 0 for ... 8 with *EachBlob* for 8 ...
EachSquare	= ZonalRanking	of *Squares*	
ThisFar	= FocalProximity	of (LocalRating	of EachSquare with 0 for 2)
ThatFar	= FocalProximity	of (LocalRating	of*EachSquare* with 0 for 1)
Centrality	= LocalSum	of *ThisFar*	and *ThatFar*
EachCircle	= LocalRating	of *BlobShape*	with *EachBlob* for ... 8 with 0 for 8 ...
HowCentral	= ZonalRanking	of (ZonalMean	of *Centrality* within *EachCircle*)
TheCircle	= LocalRating	of *HowCentral*	with 0 for 2 ...

Fig. 5. A map algebraic program. The procedure used to identify that "circle between the two squares" on the map layer sown in Fig. 1 is expressed as a sequence of map algebraic statements. Note that each statement specifies a local, zonal, focal, or incremental operation that transforms one or more existing layers (starting with one called Blobs) into a new layer (identified by a descriptive name in italics).

'Search', 'Buffer', 'EucDistance', 'Gdistance', and 'Point-To-Circle'. In other ways, however, the syntactic aspects of such a language may be of semantic and pragmatic significance as well. Should statements begin with imperative verbs or with nouns naming input(s) or output? Should objects and actions be clearly distinct, or should such distinctions be blurred? To what extent should the name of an operation suggest its functional group? And what about those statements that demand a lengthy string of modifiers? Questions of this sort become particularly important as digital cartographic models become more complex, as they attract more and more non-technical users, and as model building techniques begin to make use of sophisticated user interfaces.

5. Map algebraic origins: a checkered past

The idea of organizing and processing geographic data on a layer-by-layer basis was well developed long before the advent of modern computing. Steinitz et al. (1976) presented a historical overview of manual 'overlay mapping' techniques extending from Manning (1913) at the turn of the century to McHarg (1969) in the 1960s.

It was also in the 1960s that a computer program called SYMAP (Fisher, 1966) was developed, translating many of the same principles into digital form. By the mid-1970s, SYMAP had spawned a program called GRID (Sinton and Steinitz, 1969), and GRID in turn had spawned IMGRID (Sinton, 1977). All were raster-based systems capable of transforming and combining map layers into new layers through specified functions. Although these functions were certainly limited, not particularly well organized, and downright difficult to use, they did embody the fundamental construct of what would become map algebra.

It was in the late 1970s that that term became associated with yet another descendant of SYMAP, a rewrite of IMGRID entitled the Map Analysis Package or MAP (Tomlin, 1980). By the late 1980s, MAP would become one of the world's most widely used geographic information systems. Distributed in source code form at virtually no cost and with few restrictions on usage, MAP also spawned its own family of user enhancements, look-alike versions, all-new renditions, and functions replicated in other systems.

The operations of the original Map Analysis Package were not developed in any particular order or guided by any grand vision. Most were developed in response to immediate needs (those of the Harvard University graduate program in landscape architecture). On the other hand, some were developed not in response to needs at all, but simply because of the engaging prospect of trying to build a better mousetrap. This was a time when applications were not only searching for tools, but tools were also searching for applications.

It was really only after the fact that MAP operations were organized into groups on the basis of function. And that grouping itself evolved over time, from an initial version (Tomlin and Berry, 1979) that now seems remarkably inconsistent in hindsight, to an all-new version (Tomlin, 1983) that was much better organized, and finally to a refined version (Tomlin, 1990) that better anticipates systematic extension.

6. Map algebraic prospects: a change of pace and the pace of change

What is perhaps the most promising, and certainly the most conspicuous recent development in map algebra is the adoption of this approach by a number of prominent commercial GIS vendors. While look-and-feel varies from one system to the next, virtually all of the major players in this field are offering (or at least talking about) some form of modeling (as opposed to merely storage and retrieval or query) capabilities. After two decades of relative obscurity, safe in the towers of academia, spatial analysis has 'suddenly' been discovered in the GIS marketplace.

This is good news. Coupled with the extraordinary rate of recent advancement in virtually all fields of computing, the growing interest in GIS on the part of academics, and the slower but

nonetheless steady expansion of real-world applications, the pace of positive change in this field can only be expected to quicken. Changes in the nature and use of map algebra will probably involve all three of this approach's major components: data (variables), data processing (operations), and data-processing control (expressions).

In terms of data, one major trend for the near-term future is already under way: better integration of raster and vector formats. This is not simply a matter of providing the ability to translate conveniently from one to the other; many systems do that now. What remains is to understand better the implications of this impending marriage. How will responsibilities for different tasks be shared? Will each party retain his or her identity? And which fundamental view of the world, if either, will prevail? From the map algebraic perspective, it will be whichever of the two views is better able to accommodate the other.

Unfortunately, the relative ease with which map algebraic concepts and capabilities can be implemented in raster as opposed to vector form has led many to believe (and, in some cases, to assert in writing) that this approach is inherently and exclusively raster oriented. This is simply not so. In many vector-based systems, most of what are here called local operations have been available for years under the rubric of 'polygon overlay' techniques. Also, although zonal operations are less common in vector-based systems, these too are now beginning to appear, in part because their implementation requires little more than minor adaptation of polygon overlay algorithms. A number of incremental operations are already available in vector form as well, and these present no particular technical difficulty either. Only with the focal operations (and, in particular, those operating on travel-cost or line-of-sight neighborhoods) do challenging implementation issues arise. These do in fact present a very serious challenge, and one which may well prove to be a major factor in motivating better raster and vector integration.

Such concerns become less and less significant, however, as more and more types of non-cartographic data are routinely involved in GIS applications. As we move from maps and text to photographic images, solid models, audio material, video clips, and the variety of animation techniques now associated with digital multimedia, the issue of data integration takes on much broader proportions.

In terms of data processing as well, this sudden ability to extend our reach far beyond the world of two-dimensional maps demands a much broader prospective view of even the near-term future. Although extensions, refinements, and new applications of map algebra in areas such as feature extraction, interpolation, spatial statistics, error tracking, three-dimensional modeling, and spatial allocation can certainly be anticipated, much more fundamental changes now appear likely as well.

The motivation for these changes is already apparent: the inevitable trend toward computational procedures that are more complex, more specialized, and more likely to be conceived and programmed at higher levels of abstraction. This is a trend that can be seen in virtually all fields of computing and one that has recently begun to generate entirely new concepts of programming. Until recently, most modern computer programs have been constructed in a deductive or 'top-down' manner. Their formulation proceeds from generally defined components to more specifically defined sub-components, sub-sub-components, and so on. These programs are then implemented in 'procedural' form: as a sequence of explicitly specified processing steps.

One alternative to this style of programming is to employ inductive or 'bottom-up' techniques. These make it possible to construct large and complex programs by working on individual parts independently, while leaving the task of organizing those parts to the software compiler. Perhaps the best examples of this are in the use of 'knowledge engineering' techniques such as those associated with artificial intelligence and expert systems.

Another alternative to conventional programming methods lies in the use of nonsequential forms of execution. Among the most elementary forms of this can be seen in the conditional loop-

ing capabilities or the blurred distinction between nouns and verbs (variables and operations) in conventional programming languages. More sophisticated and more powerful examples include a variety of techniques (such as parallel distributed processing, genetic algorithms, cellular automata, and chaotic dynamics) that are associated with what have come to be called 'biocomputing' and 'object-oriented programming'. Common to all of these techniques is the idea of simulating complex phenomena not by anticipating and choreographing a program's every move, but by merely defining the general behavior of the elements involved, and then letting those elements loose to behave on their own.

In terms of data-processing control, one can also anticipate changes. On the one hand, control mechanisms are merely the packaging in which processing capabilities are delivered. On the other hand, the value of these capabilities can be realized only to the extent that their packaging lends itself to effective use. To accommodate an expanding variety of data types and data-processing methods, it is certainly likely that new channels of communication will be employed and that these will be used simultaneously. We are already seeing the benefits of highly graphic interfaces and at least beginning to explore the use of audio channels as well. We are also rapidly moving beyond the concept of a 'map' as a static object to that of a dynamic and interactive simulation.

As our ability to bring more and more realism into these simulated worlds increases, we will eventually exceed the practical utility in doing so. Long before that happens, however, a more fundamental change in our use of this technology will probably be well under way. Rather than bringing more of the real world into simulation laboratory, our objective will soon be to move in the other direction.

7. Conclusion

This paper has attempted to place in perspective the current state of a still-evolving set of tools and techniques. For those involved in this evolution, its quickening pace can be at once both exhilarating and intimidating. Both of these feelings tend to subside, however, when ends are distinguished from means. It is not the tools or techniques employed but the answers found and decisions made that will ultimately measure the merit of map algebra and the field of the GIS.

References

Fisher, H.T., 1966. SYMAP. In: Selected Projects: 1966–1970. Laboratory for Computer Graphics and Spatial Analysis, Harvard Graduate School of Design, Cambridge, MA.

Manning, W., 1913. The Billerica town plan. Landscape Archit., 3(5): 108–118.

McHarg, I.L., 1969. Design with Nature. Natural History Press, New York.

Sinton, D.F., 1977. The Users' Guide to IMGRID: An Information Manipulation System for Grid Cell Data Structures. Department of Landscape Architecture, Harvard Graduate School of Design, Cambridge, MA.

Sinton, D.F. and Steinitz, C.F., 1969. GRID: A Users' Manual. Laboratory for Computer Graphics and Spatial Analysis, Harvard Graduate School of Design, Cambridge, MA.

Steinitz, C.F., Parker, P. and Jordan, L., 1976. Hand-drawn overlays: their history and prospective uses. Landscape Archit., 66(8): 444–455.

Tomlin, C.D., 1980. The Map Analysis Package. School of Forestry and Environmental Studies, Yale University, New Haven, CT.

Tomlin, C.D., 1983. Digital cartographic modeling techniques in environmental planning. Unpublished doctoral dissertation, Graduate School Division of Forestry and Environmental Studies, Yale University.

Tomlin, C.D., 1990. Geographic Information Systems and Cartographic Modeling. Prentice-Hall, Englewood Cliffs, NJ.

Tomlin, C.D., 1991. Cartographic modeling. In: M. Goodchild, D. Maguire and D. Rhind (Editors), Geographical Information Systems: Principles and Applications. Longman, Harlow, UK, Chapter 23.

Tomlin, C.D. and Berry, J.K., 1979. A mathematical structure for cartographic modeling in environmental analysis. In: Proceedings of the 39th Symposium of the American Congress on Surveying and Mapping, 18–24 March 1979, Washington, DC. American Congress on Surveying and Mapping, Falls Church, VA, pp. 269–283.