Integrating Dynamic Spatial Models with Discrete Event Simulation

David Pullar
Department of Geographical Sciences
The University of Queensland
BRISBANE, QLD. 4072

Keywords:
Environmental, Geographical Information Systems, Spatial Modeling

ABSTRACT: Dynamic spatial modeling addresses computational aspects of space-time process modeling. The types of processes it is meant to deal with are commonly found in Earth and biological sciences, such as hydrology and ecology. Geographical information systems (GIS) are a popular tool for modeling these application areas. A GIS provides spatial data structures, query capability and functions to manipulate spatially referenced information. However these tools currently lack any capability to perform dynamic modeling. This paper describes a prototype for a spatial modeling component that may be incorporated in GIS and computer simulation software. The component features a language and spatial operators suited to discrete-time analysis. In particular it describes control constructs to analyse local spatial neighborhoods. A model for hydrological surface water runoff is described as an example application.

1. Introduction

Dynamic spatial modeling addresses computational aspects of space-time process modeling. The types of processes it is meant to deal with are commonly found in Earth and biological sciences, such as hydrology and ecology. Geographical information systems (GIS) are a popular tool for modeling these application areas. A GIS provides spatial data structures, query capability and functions to manipulate spatially referenced information. However these tools are currently lacking in capability to perform dynamic modeling. Some recognized drawbacks are [1]:
(i) mismatch between the view of reality of physical modelers and GIS data models,
(ii) data types do not adequately model real world variation,
(iii) hard wired models are too inflexible to be modified,
(iv) writing a model often requires advanced skills in computer programming,
(v) visualization of time sequenced spatial processes are not supported, and
(vi) no support for processes spatial models, i.e. reactive diffusion, cellular automata.

While one could attempt to address these shortcomings individually in a GIS, a more sensible approach is to use an interactive simulation environment for building and analysing dynamic spatial models. We believe this can be achieved by developing a dynamic spatial component as a specialist library block that integrates into existing simulation systems. The component understands its own interpretation of a state-space variable as a spatially referenced surface. Spatial manipulation of this surface would of course require a spatial algebraic language that understands surface forms. The next section describes the nature of such a language. A subsequent section will discuss the main language features that are needed to support spatial dynamic simulation. One of the more important aspects of this modeling is to be able to perform neighborhood analysis over a surface in a controlled manner. For instance, the way advective-diffusive processes propagate across surface landforms. Our implementation of the spatial algebra is then described. The implementation is called MapScript; it is an ActiveX component that processes inputs according to a lightweight scripting language. The language provides basic scientific calculator functionality and computational control for multi-component surfaces. Generally, scripts perform one time step calculation within a larger model that is run within a simulation program. The final section demonstrates the use of the scripting language and simulation model for a hydrological application.

2. Spatial Models

Geographical phenomena can be conceptualised from two perspectives: either as a continuously varying field, or as a set of discretely bounded objects [2]. This is similar to the field and particle views in quantum mechanics. A spatial object representation is suited to describing properties and relationships between identifiable spatial objects. While there are many applications that require this type of spatial model, it is generally inappropriate for environmental information. This is because the natural environment is characterized by continuous variation and dynamic change, rather than well-defined static objects. The field model, on the other hand, does provide an appropriate representation of environmental phenomena. An univariate or multivariate field model can represent the continuous variation and complexity of an environmental surface. There are no exact mathematical properties to describe the behavior of point attributes and their neighboring values, but knowledge of environmental data deems that values will have a smooth pattern of variation across space. This can be modeled statistically to account for random variations as a spatially correlated variable. This pattern of variation and the factors to which it
relates form the basis for a simplified model of complex environmental data.

An image-based data structure geo-referenced to the Earth's surface provides a computer representation of simple surfaces, and a multi-image can be used for more complex multivariate data. Processing follows the same conventions for digital image processing; namely point, local, and geometric operations [3]. A popular conceptual basis for image-based spatial analysis is called map algebra [4]. Map algebra uses a function-oriented language to operate on four implicit spatial data types: local, neighborhood, zonal, and whole surface scenes. It is used for tasks ranging from land suitability modeling to mineral exploration in the geosciences [5]. As an example, site suitability for a housing construction needs to look at the land stability for soils and geology. Soil and bedrock geology map data are reclassified to engineering suitability ratings (1 most suitable to 5 least suitable) and summed for an average over the site. The map algebra statements for this model are given below.

\[
\text{soil-suit} = \text{reclass}(\text{soil} = \text{clay}, 2), \text{soil} = \text{sandy}, 3), \text{soil} = \text{black}, 5), \ldots).
\]

\[
\text{bedrock-suit} = \text{reclass}(\text{geology} = \text{sandy-shale}, 4), \text{geology} = \text{granite}, 1), \ldots).
\]

\[
\text{out-suit} = \text{soil-suit} + \text{bedrock-suit}.
\]

\[
\text{site-suit} = \text{zonalmean}(\text{site}, \text{out-suit})
\]

In the example above, \text{reclass} is a local operator that operates on each pixel location in an image, and \text{zonalmean} is a zonal operator that operates on a group of pixel's that fall within a given area. Because these data types are implicitly represented in an image layer this opens the possibility for having other interpreted forms.

Map algebra provides a high-level language interface to describe and manipulate surface data. Common examples of surface data include terrain models, categorical land cover maps, and scalar temperature surfaces. Appropriate definitions and algebraic operations can be provided for these types of surface representations. This permits users to manipulate data at a level closer to their understanding of the information as opposed to performing sets of low level operations on image data structures. All the intelligence to express models is then vested in the map algebra models, and not hard coded into software source code.

3. Simulation and Spatial Models

Spatial models fit nicely with simulation modeling concepts. Given a deterministic system \(\langle T, U, Y, Q, \Omega, \delta, \lambda \rangle\) where:

- \(T\) is time set
- \(U\) & \(Y\) are input and output sets
- \(Q\) is state-space of system
- \(\Omega\) input functions, \(Q \times \Omega \rightarrow Q\)
- \(\delta\) transition functions, \(Q \times T \times Q \rightarrow Q\)
- \(\lambda\) is the output function.

The state-space of the system \(Q\) is often expressed as a function of \(t\) in \(T\). Changes in the state-space occur continuously as \(\partial u/\partial t\), and discretely at events occurring because of discontinuities in transition functions or because of explicit events defined by rules and conditions. In spatial modeling the state-space is a lattice composed of scalar or categorical values along a two dimensional field \(X\). Changes in state-space occur continuously as \(\partial u/\partial t + \partial u/\partial x\), and discretely at events occurring because of discontinuities in transition functions or because of explicit events defined by boundary rules.

Any spatial models involve convoluting a template over an entire lattice. The concept is generalized in image processing as image algebra [6], and in cellular automata using patterns [7] or morphological structures [8]. The convolution operation performs a sweeping process invoking a calculation on the cells within the support region for a template. A subset of cells \(X\) is mapped to a new set \(Y\) and some a measurement is made of \(Y\), i.e. \(y' = \mu(\Psi(x))\). Measurement calculations include performing interpolation, aggregation, filtering values, derivative calculations, contiguity assessment, dilation, and evaluating surface characteristics.

While this type of analysis is also applicable to spatial models, there still remains a large class of environmental models that do not obey this clockwork type of processing. Processes that involve spreading or transport act along environmental gradients within the landscape. Therefore special control needs to be exercised on how the convolution operation is performed. Burrough's [1] describes two extra control mechanisms for diffusion and directed topology. Figure 1 shows the three principle types of processing orders, and they are explained below:

- i) row scan order governed by the clockwork lattice structure,
- ii) spread order governed by the spreading or scattering of a material from a more concentrated region,
- iii) flow order governed by advection which is the transport of a material due to velocity of as a medium.
Dynamic models provide the additional dimension where the cell values change over time. Dynamic modeling languages in GIS have been developed using global functions for diffuse and advective processes [9]. However, global operators tend to be application specific. We propose a modeling language with special iteration constructs that combine process control and the flexibility of convolution operators. The next section describes an implementation of such a language.

4. MapScript Spatial Modeling Language

MapScript is a lightweight language for processing image-based GIS data using map algebra. The language parser and engine are built as a software component to interoperate with existing GIS or simulation software. Currently MapScript can be run with a GIS system called IDRISI [10], and data flow simulation software called MathConnex [11].

MapScript is built in C++ with a class hierarchy based upon a value type. Variants for value types include numerical, boolean, template, cells, or an image. Determination of the type for a script variable is performed at run-time. MapScript supports basic arithmetic and relational comparison operators. The three basic language constructs of MapScript are: a functional assignment statement, iteration operator (docell, doflow, dospread) to iterate over cells in an image map, and an if..else..end conditional control to direct program flow. Templates are supported with any arbitrary size and designation of a center. Basic operations on templates are also supported in a similar fashion to Ritter et al [6], with the addition of template operations in cell iterations. Algebra operations on templates and a processing cell result in a new template, which is measured and assigned to a new processing cell. This is the typical way new cell values are computed from neighboring cells using morphological operators, i.e. \( y' = \mu(\Psi(x)) \). Further details of the language are given in the appendix and the next section provides an example of a program script.

5. Hydrological Runoff Example

This section presents an example of a model for the distribution of rainfall runoff within a hydrological application. It is an interesting example because of the complexity and variation in the behavior of water as it interacts with the land surface. Given a rainfall event we model the movement of water over a surface as it accumulates and eventually drains away. Initially rainfall infiltrates into the ground, but as the ground becomes saturated water will begin to accumulate and form excess flow. This flow is influenced by terrain properties such as surface roughness, slope, and geometry. The distribution of accumulated rainfall excess and flow characteristics involves a lagged time relationship, namely the time it takes for water to move across the surface. This is expressed in two equations. First, the storage relation describes changes in accumulated rainfall with respect to the difference between inflow and outflow for a cell. The hydrologic continuity equation for a system is:

\[
1 - Q = \frac{dS}{dt}
\]

where:

\( I \) = inflow in vol/time
\( Q \) = outflow in vol/time
\( dS/dt \) = change in storage in vol/time.

Inflow is a combination of flow received from the uphill adjoining cell and from rainfall excess. Flow is a dynamic process that occurs in space and time. Assuming steady uniform flow conditions, equations for momentum that relate discharge, runoff velocity, and the water surface depth to time are given by kinematic wave equation as:

\[
\frac{\partial y}{\partial t} + \frac{\partial q}{\partial x} = i_e
\]

where:

\( y(x,t) \) is the depth of overland flow
\( q(x,t) \) is the rate of overland flow / unit width
\( i_e \) is the net rainfall rate (i.e. precipitation – infiltration)
Of course the rate of overland flow may also be expressed in terms of a velocity function. Manning's equation for overland flow is adapted as follows:

\[ q = \frac{vy^{5/3}}{N} \cdot \sqrt{s} \cdot y^{5/3} \]  

where:
- \( N \) is effective roughness parameters for overland flow
- \( s \) is the average flow slope
- \( v \) is the dynamic wave velocity.

Finite-element approximations are made to solve the above partial differential equations. Bedient and Huber [12] describe the calculations and provide a simple program for the one-dimensional case of computing flow along a strip of unit width. All calculations progress from an uphill cell to the downslope cell. Flow distance is measured in cell size \( \Delta x \) per unit width. One strip is processed during a time interval \( \Delta t \). See figure 2.

The conservative solution to equation 2 is found as follows:

\[ q(x + \Delta x, t + \Delta t) = q(x, t + \Delta t) + \Delta x \cdot q'(x, t + \Delta t) \] (Eq. 4)

Solving for \( q' \) using a backward-difference approximation of Taylor's series expansion where second and higher-order terms are ignored gives:

\[ q(x + \Delta x, t + \Delta t) = q(x, t + \Delta t) + \Delta x \cdot \frac{y(x, t + \Delta t) - y(x, t)}{\Delta t} \] (Eq. 5).

Now re-arranging terms in equation 3 we solve for flow depth:

\[ y(x + \Delta x, t + \Delta t) = \left( \frac{q(x + \Delta x, t + \Delta t)}{v} \right)^{\frac{3}{5}} \]  

(Eq. 6).

Figure 2: Computation of current cell \((x + \Delta x, t + \Delta t)\)

The one-dimensional case is relatively easy to solve. The two-dimensional case for a surface is also simple to solve, but the data handling to deal with multiple flow paths makes it complex to express in software code. Johnson and Mill [13] describe an implementation of this example using a raster data structure programmed in the C language. The length of the program is significantly longer and more complex than the programs given in Bedient and Huber [12]. Note that the solution described is the conservative method, in practice one would use different methods depending upon the numerical stability of the solution. This is related to the distance step \( \Delta x \) and time step \( \Delta t \) used in calculations.

The map script for this conservative solution is as follows:

**Desc:** conservative finite-difference runoff calculation for kinematic wave

**Input:** rain_excess, flow, flow_velocity

**Output:** discharge
doflow (terrain)

\[ \text{discharge} = \text{sum(inflows(discharge))} + \rightarrow \text{sum(inflows(depth) - inflows(prev_depth))} \]

\[ \text{depth} = \text{pow(discharge/velocity, 5/3)} \]

endflow

Figure 3: Simulation of hydrological model for rainfall runoff.

The script uses a doflow iteration to process cells from the cell with the highest elevation to the cell with the lowest elevation. The inflows() operation produces a template for neighboring cells that have contributing flows, which is subsequently evaluated using an aggregate operation. Figure 3 shows a block-diagram model for the complete simulation. Each block is a component in the system model. The simulation is driven by a storm event that lists rain series data for different regions. Flow velocities are calculated and remain the same throughout the simulation. The runoff model computes discharge, which is visualized as an animated cartographic map using a special mapping component. The model halts when the times series data
in the storm events ends, or alternatively if there is no discharge from a catchment.

5. Conclusion

This paper describes a map algebra language that allows modelers to easily construct spatial simulation models. The language addresses many of the shortcomings found in existing GIS's. Namely it provide modelers with a high level view of the problem using a data model that has a intuitive representation of land phenomena. The modeling language provides flexibility for writing scripts to model physical land processes and can be combined into a simulation system for interactive development of applications and visualisations of results.

Appendix

MapScript may be obtained on the Internet at: http://www.geosp.uq.edu.au/projects/MapScript/.

6. References


7. Author Biography

David Pullar is a lecturer in spatial information science at The University of Queensland. He has worked in industry with Environmental Systems Research Institute in Redlands, CA. And as a University Research Assistant with the National Center for Geographic Informations Analysis (NCGIA) sponsored by the National Science Foundation in the US. He received his Ph.D. from the University of Maine, US in 1994.