

Towards A Unified Framework For Spatial Data Models

Gilberto Câmara, Antônio Miguel Vieira Monteiro and João Argemiro Paiva
INPE — Instituto Nacional de Pesquisas Espaciais, Caixa Postal 515
12201 São José dos Campos, SP, Brazil
{gilberto,miguel, miro}@dpi.inpe.br

Jonas Gomes and Luiz Velho
IMPA- Instituto de Matemática Pura e Aplicada,
Estrada Dona Castorina, 11
22460 Rio de Janeiro, RJ, Brazil
{jonas,lvelho}@impa.br

Abstract *This paper describes a unified framework for the problems of modelling and processing spatial entities. We propose a general definition of **spatial objects**, and show that the different types of spatial data can be expressed as particular cases of this definition. Furthermore, we present a taxonomy for the various types of GIS operations, defined in terms of the properties of this definition. Our goal is to argue that GIS data types and operations can be defined based on a single formal notion, which encapsulates the GIS concepts of both continuous fields and discrete features, with important consequences for system and interface design, interoperability issues and language proposal.*

Keywords: *Geographical Information Systems, Spatial Databases, Geographical Data Models.*

1 Introduction

This paper discusses the problems of modeling geographical reality into the computer and proposes a unified formal definition for spatial data sets. To begin with, we must establish the conceptual framework in which our definitions are proposed and relate our vision to the general perspectives of spatial data modeling.

In this paper, we consider that modeling of geographical reality into a GIS (geographical information system) is a process which includes:

- An informal description of the real-world entities which will be represented in the computer, and of the spatial and non-spatial relations between these entities.
- The mapping of these entities to one or more of the geometrical representations available in a GIS.

The choice of the geometrical representations is dictated by the expected results to be obtained and by considerations such as sampling and interpolation error, end-

user presentation and ease of performing query and manipulation operations. Whilst this situation is true in general in any information system, these decisions are crucially important in the case of spatial data sets, given the multitude of geometrical data structures and query and manipulation functions available.

One of the most important concerns is whether to choose a *feature-based* or a *field-based* model for the spatial entities [1]. The *field model* represents spatial data as a set of continuous distributions on a two-dimensional support. The *feature model* represents the world as a surface occupied by discrete, identifiable entities.

As some authors have already pointed out [2], the *field* and *feature* views have an underlying common notion, which is the implicit reliance on *cartesian* (or *absolute*) space as a priori frame of reference for locating spatial phenomena. In this view, cartesian space is simply a neutral container within which all physical processes occur. The primitive notion on a cartesian space is the idea of georeferenced location. Each entity of space is associated to one or more locations on the Earth, and the spatial relations are derived from the location.

The alternative to absolute space is to consider a *relative* notion of space [3], constituted through the spatial relations arising among geographical entities. In the framework of relative space, the primitive notion is that of spatial relation between entities. Spatial models such as spatial interaction models and location-allocation models used in transportation are examples of application which use the *relative* notion of space.

Current GIS technology embodies an absolute view of space, since the most common geometrical representations available in GIS – grids, TINs, planar vector maps – are all based on the notion of a georeferenced location. It is therefore not surprising that the notions of features and fields – as defined in the current GIS literature – can be generalized into a single formal definition, as presented below.

Based on the concepts of fields and features, and operating within the framework of an absolute space, we propose a general formal definition of *spatial objects*. We also show that the different types of spatial data used in a GIS can be expressed as particular cases of this definition. Furthermore, we present taxonomy for the various types of GIS operations, in terms of properties of this definition. Our goal is to show that GIS data types and operations (used within the cartesian space framework) can be defined based on a single formal notion, with important consequences for system and interface design, interoperability issues and language proposal.

The remainder of the paper is organized as follows. Section 2 introduces the abstraction paradigm used as a basis for our concepts, and considers the general problem of computer representation of geographical reality. Section 3 proposes a general definition for spatial objects, discusses the representation of these objects and gives a classification. Section 4 summarizes the typical operations on spatial data, as seen in the perspective of our framework. Section 5 outlines some consequences of this approach, both from a theoretical and a practical point of view.

2 Abstraction Paradigms For Spatial Modeling

Spatial data modeling for GIS can be viewed as a special case of computational modeling of physical phenomena. In computational modeling, a very important and necessary step is to establish mathematical models, which create the abstract descriptions of the real-world entities of interest. In a classical paper in the area of geometric modeling, Requicha [4] established a conceptual framework which distinguishes between the physical, mathematical and representational levels of abstraction. His work was further extended by Gomes and Velho [5] which propose the “four universes paradigm” as a general modeling mechanism for applied computational mathematics in general, and not only for geometric modeling. The four abstraction levels are described as (See Figure 1):

- The physical universe, which comprises the real-world entities that will be modeled in the computer.
- The mathematical universe, which includes a formal definition of the entities which are included in the model.
- The representation universe, which defines how the various continuous models are discretized.
- The implementation universe, where the data structures are associated to the discretized objects of the representation universe.

These abstraction levels have been successfully used in different areas of computer graphics. An extensive use of the paradigm is found in [6], where it has enabled the formulation of a robust conceptual approach both for theoretical issues and implementation, related with the subject of warping and morphing.

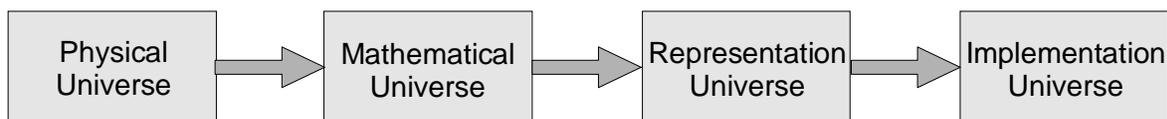


Figure 1 – Abstraction levels in Computer Modeling (from Gomes and Velho, 1995)

Our experience has shown that the use of the four universes paradigm for spatial data, is also particularly suited: At the physical universe level, we find elements such as *parcels, rivers and soil maps*; at the mathematical level, we define *spatial objects*, and specialize this notion into classes of geographical data which encompasses traditional concepts from GIS such as *fields and features*; at this level, we distinguish between *raster* and *vector* representations, which may be further specialized, such as *grids, TINs, image* structures for raster and *arc-node* and *arc-node-polygon* structures; at the implementation level the actual coding takes place and we find data structures such as *R-trees* and *quad-trees*.

One of the consequences of the multiple levels of abstraction paradigm is to make an explicit reference to design decisions, which are very often implicitly and informally defined. For example, the use of raster data structures to store terrain models in a computer implies a mathematical formulation of what is a terrain. In this approach, we are compelled to define the elements of each universe and to establish the relation between the elements of each universe.

3 Spatial Objects

Using the abstraction paradigm introduced in the previous section, the most important issue is concerned in characterizing the mathematical entities that will describe the elements from the physical world. In the case of geographical data, this leads us to the concept of a *spatial object* (or simply *so*) as the basic element of the mathematical universe. A spatial object is a triple $so = (S, A, f)$ where:

1. $S \subset \mathbb{R}^2$ is a subset of the Euclidean plane, it is the *geometrical support*.
2. A is a set of *attribute domains* A_1, \dots, A_n ,
3. $f: S \rightarrow A_1 \times A_2 \times \dots \times A_n$ is the *attribute function* of the spatial object, which associates, to each location in the support, a value on the set of attribute domains.

Note that this definition follows from that of a *graphical object* introduced in [7]. It caters to the different subtypes of geographical data. We should remark that this definition, in accordance to our paradigm, is completely generic and is not bound to any particular representation.

A spatial object is *geo-referenced* if there exists a parameterization g from the geometrical support S to the surface of the earth. Mathematically the map g can be described approximately by a parameterization from S to the surface of a sphere. Different parameterizations are possible

related with the various map projections of the sphere.

3.1 Representation of Spatial Objects

Once we have a spatial object the first step towards implementation is to obtain its representation. The representation of a spatial object consists in discretising both its geometric support and the attribute function. The representation of the geometric support consists in representing its topology and geometry. This topic is extensively studied in the area of geometric modeling. In GIS it is used mainly the representation techniques based on decomposition. These techniques employ a top-down methodology where the geometric support is decomposed into simpler geometrical objects that are easier to represent.

As an example, consider a very important type of geographical data: a *terrain*. As a spatial object a terrain is a real valued function $f: S \rightarrow \mathbb{R}$, where S is a subset of the euclidean plane. Two commonly used representations for terrains are the *TIN* (Triangulated Irregular Network) and the *grid*. A TIN representation subdivides the geometrical support S of the terrain into triangles in such a way to form a triangulation (intersecting triangles should share a vertex or edge). A grid representation subdivides the geometrical support S into small rectangles so as to construct a lattice. Figures 2(a) and 2(b) show, respectively, a TIN and a grid representation of a terrain.

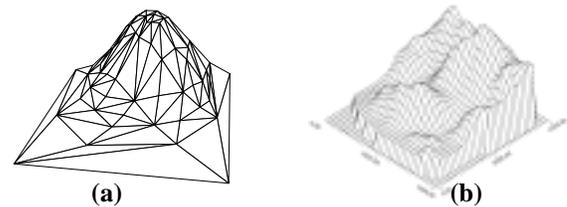


Figure 2 – TIN (a) and Grid representation (b) of terrain.

If we are interested only in the geometry and topology of the terrain it is enough to use a TIN or a grid representation. Nevertheless, in order to obtain a complete representation of the spatial object we must represent the attribute function on the TIN or on the grid. Sampling the function at each element of the representation generally attains this. On a TIN samples are generally taken at the vertices of the triangles; on a grid samples can be taken either at the vertices of the grid or at the center of each rectangular cell.

Note that because the grid decomposition can be easily structured into a rectangular lattice, this representation is completely characterized by giving the number $m \times n$ of

decomposition cells (rectangles) and the value of the attribute function in each cell. The structuring is given by the natural row/columns ordering of the lattice. The grid representation of a spatial object is called a *matrix* or *raster* representation. The order $m \times n$ of the decomposition matrix is called the spatial resolution of the representation.

To conclude the representation of the spatial object the attribute function should assume only a finite number of attribute values (this corresponds to the use of a specified number of bits to represent the attribute values). The process of discretizing the attribute function is called *quantization* (this comes from the usage of the term in the area of image processing). Each of the values assumed by the function is called a *quantization level*. The number of quantization levels to be used depends on the nature of the geographic data being represented.

We should remark that if the attribute function is quantized to n levels, a_1, \dots, a_n , then it determines a partition of the geometric support of the spatial object into a collection of disjoint sets $S_i, i=1, \dots, n$, such that the attribute f at each set S_i is constant and equal to a_i . This is illustrated in the Figure 3 below.

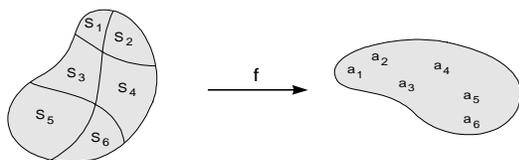


Figure 3 – Attribute function with n values and associated partition

An important issue when representing spatial objects consists in reconstructing the object from its discrete representation. We will not discuss this topic in this article.

3.2 Classification of Spatial Objects

Spatial objects are classified according to the topology of the support, and the values of the attribute function. Four different classes are possible: simple, composite, homogeneous and non-homogeneous spatial objects.

A spatial object is called *simple* if its support S is a connected region in \mathbb{R}^2 . Otherwise it is called a *composite* object. A spatial object is called *homogeneous* if its attribute function assumes a constant value $f(s) = (a_1, \dots, a_n), \forall s \in S$. Otherwise the spatial object is called *non-homogeneous*. Some examples will be given below.

3.2.1 Simple spatial objects

Figure 4 shows an example of a simple spatial object.

The object describes a country named “Brazil”. The geometric support S is a connected region, and it has two constant attribute values: “name” and “population”.

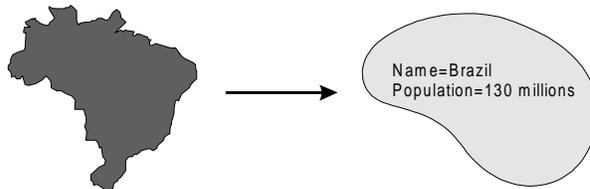


Figure 4 - A simple and homogeneous spatial object.

In practice, the notion of a *single, homogeneous spatial object* is too simplistic and we need to use the more complex classes of spatial objects to characterize the geographical entities.

3.2.2 Composite homogeneous spatial object

In this case, we are dealing with a geographical object where the geometric support S has several connected components and the attribute function assumes a constant value. Figure 5 shows an example, which describes the country named “Japan”, where S is a set with four connected components (representing the main islands of the Japanese archipelago), and the attribute domains A_1 and A_2 are “name” and “population”

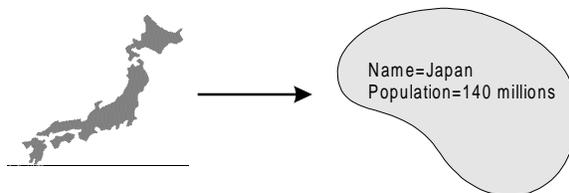


Figure 5 - Example Of A Composite Homogeneous Object.

3.2.3 Simple non-homogeneous spatial object

For these object types, the support S is composed of a connected region, but the attribute function f varies for each point in the region. Depending on the number of levels used in the representation of the attribute function, the geometric support is partitioned into a finite number of sets. This concept corresponds to the notion of fields [2], such as vegetation maps and topography. Figure 6 shows an example of a simple non-homogeneous object (a vegetation map). Note the partitioning of the geometric support as we described in Figure 3.

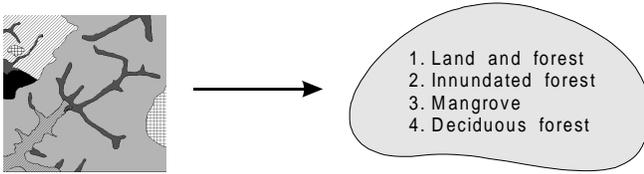


Figure 6 - Example of a simple and non-homogenous object.

3.2.4 Composite non-homogeneous spatial object

In this case, the geometrical support S is composed of several connected components, and the attribute function is not constant over S . This definition, in its generality, has no direct counterpart in the traditional GIS elements. However, there is a very important special case, namely:

- The geometric support is a disjoint collection of connected regions, $S = \cup S_i, S_i \cap S_j = \emptyset (i \neq j)$;
- The attribute function f is constant on each set S_i , and the constant values assumed are distinct, in general.

This special type of composite non-homogeneous spatial object constitutes a formal definition of the *layers* of a GIS, such as the notion of *coverage* in the well-known ARC/INFO™ system [8].

Figure 7 shows an example of a layer, which corresponds to a map of the South Asia region, composed of different islands. The map is dealt by the system as a single object, whose components (the geometrical supports associated to each island) are mapped to different values of the attribute set.

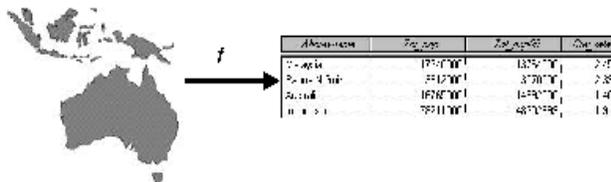


Figure 7 - Example of a composite, non-homogeneous object.

3.3 Relation to GIS Literature

In this section, we will attempt to relate the concepts introduced above with the traditional concepts in the GIS literature, especially those used by the OpenGIS® consortium [9]. Since the latter definitions are mostly informal, an exact matching is not always possible, but the approximate correspondences are already illustrative of the

hazards of semantic model conversion between systems.

The OpenGIS® model is based on an abstract class (*feature*) which has two specializations: *feature with geometry* and *coverage*.

The definition of *feature with geometry* allows for complex geometrical representations to be associated to the same feature and for different features to share the same geometrical representation. The notion of *feature with geometry* in OpenGIS® corresponds roughly to our definition of a single non-homogeneous spatial object (Section 3.2.2).

An OpenGIS® *coverage* is an association between a geometric description of entities and a set of attributes; this association is defined by a coverage function (or *c_function*) $f:(geometry) \rightarrow (attribute\ set)$. The OpenGIS® proponents chose not to use a representation-independent definition for *coverage*, and to define it in terms of specific geometrical representations, such as a *grid coverage*, *TIN coverage* or *geometry coverage* [9]. Most of the OpenGIS® specializations of coverages, such as *sample*, *grid*, *TIN* and *image* can be considered as computer representations of the concept of a *simple non-homogeneous object* (Section 3.2.3), using specific data structures. However, their definition of *geometry coverage* can be considered as a special case of a *composite non-homogeneous object* (Section 3.2.4).

Thus, we can observe that the definitions of *coverage* on the OpenGIS are based on different formal concepts. In other words, a *grid coverage* in OpenGIS does not have the same formal basis as a *geometry coverage*. This situation may lead to potential problems in understanding and using this concept as a basis for interoperability. In fact, in a previous work [10], we have argued against the choice of the OpenGIS consortium of using industry terminology, such as *feature* and *coverage*, which is already content-rich and are associated by the users with existing semantic concepts.

Table 1 provides a resume of the relation between our definitions and established industry notions.

TABLE 1 - CORRESPONDENCE OF DEFINITIONS

	Connected geometrical support	Non-Connected geometrical support
Homogeneous		<i>Simple feature (OpenGIS)</i>
Non-homogeneous	<i>Tin, Grid, Image and SurfaceCoverage (OpenGIS)</i>	<i>Coverage (ARC/INFO), Geometry Coverage (OpenGIS)</i>

4 Operations on Spatial Data

One of the main purposes of a single formal definition for spatial objects is to provide a unified perspective on the semantics of GIS operations. In current practice, the use of different models for fields and features usually leads to implementation of different subsystems on a GIS: raster map operations implementing Tomlin's Map Algebra [11] and vector spatial queries with languages similar to Spatial SQL [12]. This situation is not always desirable. For example, "overlap" is a well-known concept for denoting a topological configuration between two geographical entities [13]. In many GIS, it is possible to inquire if two entities represented by vector polygons "overlap", but it is not straightforward to inquire if the same property holds for two digital terrain models.

4.1 Atomic Operations on Spatial Objects

A general GIS should have basic operations, which are generally applicable to different types of geographical entities, including both *field* and *feature* models. Therefore, we have chosen to characterize what we propose to be a minimal set of atomic operations, applicable to different types of spatial objects (described in Section 3). Upon this minimal set, more complex operations can be applied, eventually leading to more specific operation classes such as Map Algebra. In this section, we propose such a minimal set and in the next one, indicate how these atomic operations can be used as building blocks of more complex ones.

We consider three classes of operations with spatial objects: Attribute-based, Spatial-based and Creation-Delete operations. We will describe these operations below.

In keeping to our paradigm, which distinguishes between the mathematical and the representation universes, our definition of operations on geographical objects does not consider representation issues. These issues are considered to be implementation-dependent. For example, the practical implementation of a mathematical operation between spatial objects may require their conversion from a vector to a raster representation. Ideally, this conversion should be performed automatically by the system (under certain rules). The fact that most commercial systems require the user to request an explicit vector-to-raster conversion only serves as indication that GIS technology would benefit from a formal basis, much as the database technology was improved by the relational model.

Using a formal definition as a basis, we can distinguish which operations are essential part of the properties of geographical objects and those that are constrained by representation-based issues.

4.1.1 Attribute-based operations

Two basic operations are proposed: VALUE (λ, A_i), which returns the value of attribute A_i at location λ (this operation is actually the value of the attribute function for attribute A_i), and ASSIGN (λ, A_i, a), which assigns a value a for attribute A_i at location λ . For shorthand notation, we indicate these operations as $f(\lambda, A_i)$ and $\alpha(\lambda, A_i, a)$, respectively.

4.1.2 Spatial-based operations

Spatial based operations can be further classified based on a spatial predicate denoted by ξ . As an example we have the following cases:

a) Topological restrictions

Given a pair of spatial objects so_i and so_j , and a topological predicate θ , these operations return a boolean value ($\{\text{true}, \text{false}\}$) based on the result of the application of the topological predicate $\theta(so_i, so_j)$. The topological predicates proposed are EQUAL, DISJOINT, INTERSECT, TOUCH, CROSS, OVERLAP, CONTAINS, WITHIN, and RELATE, as defined by the OpenGIS consortium [9].

b) Distance-based operations

Given a pair of spatial objects so_i and so_j , the DISTANCE(so_i, so_j) operation returns the distance measure between these two objects. For shorthand notation, we indicate this operation as $\text{dist}(so_i, so_j)$. We can also consider a boolean predicate $\delta(so_i, so_j, d)$ which returns true or false depending whether the distance between so_i and so_j is smaller than a given value d .

c) Direction-based operations

Direction relationships between objects can be qualitatively described using the notion of cardinal directions [14]. The object support is abstracted using its minimum bounding rectangle, and the direction relationships are expressed through a 3x3 matrix representing true or false values for NORTHWEST, NORTH, NORTHEAST, WEST, CENTER, EAST, SOUTHWEST, SOUTH, and SOUTHEAST relations. Given a pair of spatial objects so_i and so_j , the DIRECTION(so_i, so_j) operation returns the cardinal direction matrix between these two objects. We can also consider a boolean predicate PATH($so_i, so_j, \text{dir_rel}$) which returns true or false depending on the direction relation dir_rel value between so_i and so_j .

4.1.3 Combined spatial and attribute-based operations

Given a pair of spatial objects so_i and so_j , this class of operation return a value which is calculated on the attribute values of so_i , based on the geometrical support of so_j . We call this types of operation **REGION**, and consider specializations such as **REGION_MAX** (A_i, so_i, so_j) and **REGION_AVE** (A_i, so_i, so_j), which compute, respectively, the maximum and the average value of the attribute A_i of the spatial object so_i , constrained by the geometrical support of the spatial object so_j . Figure 8 shows an example of a "region" operation, where the attribute of first spatial object is a numerical value, and we compute the maximum value of the region indicated by S_2 .

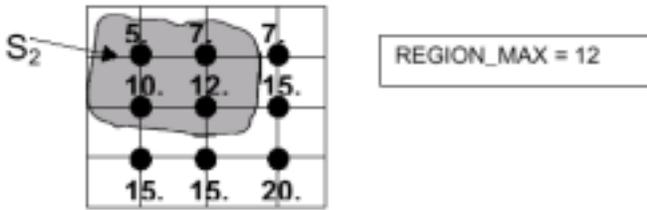


Figure 8 – Example of a region operation.

Again, it should be stressed that some operations need a conversion between representations of spatial objects (such as the conversion from a grid to a TIN). In this case the operation depends on the conversion technique used.

4.2 Composite Operations on Spatial Objects

In this section, we indicate how the usual operations of spatial queries and map algebra can be expressed in terms of the atomic operations proposed. We will consider three such types of operations: spatial selection, spatial joins and point operations. Other operations on spatial objects can be defined in a similar fashion.

Definition 1. Spatial Selection.

The *spatial selection* operation can be defined as follows. Given a set of spatial objects $SO = \{so_1, \dots, so_n\}$, a reference spatial object so^* , the *spatial selection* operation $\varphi: SO \rightarrow SO$, given a spatial predicate ξ which relates the spatial objects $so \in SO$ to so^* is defined by:

$$\varphi_{\xi}(SO) = \{ so \in GO \mid \xi((so, so^*)) \}. \quad \square$$

The *spatial selection* operator is such that the output is a subset of the original set, composed of all spatial objects that satisfy the geometrical predicate, as in the example: “select all regions of France which are adjacent to the Midi-Pyrénées regions (which contains the city of

Toulouse)”, illustrated in Figure 9.



Figure 9 – Example of spatial selection operation

Definition 2. Spatial Join.

Let SO_1 and SO_2 be two sets of spatial objects. Let ξ be a spatial predicate computable for every pair of objects (so_1, so_2) where $so_1 \in GO_1$ and $so_2 \in GO_2$. The spatial join operation θ :

$SO_1 \times SO_2 \rightarrow SO_1 \times SO_2$ is such that:

$$\theta_{\xi}(SO_1, SO_2) = \{ (so_1, so_2) \in (SO_1, SO_2) \mid \xi(so_1, so_2) \}. \quad \square$$

Spatial join is an operation where a comparison between two sets of spatial objects SO_1 and SO_2 takes place, based on a spatial predicate which is computed over the geometrical support of these sets. The name “spatial join” is employed by analogy to the join operation in relational algebra. The result of the spatial join operation is a *set of object-pairs*, which satisfy the spatial restriction. One example of spatial join would be: “Find all native reservations located closer than 50 km to the main roads in Amazonia”. The answer is a set of pairs of spatial objects (reservation, road). We can also consider operations where an output is generated, based on one or more inputs.

Definition 3. Point operations.

Let so_1, so_2, \dots, so_n be spatial objects used as input, so_{n+1} be the output spatial object, A^1, A^2, \dots, A^{n+1} be the attribute sets associated to these spatial objects and S_o be the geometric support of the output. Let A_i^k be the i -th attribute of the k -th spatial object. The *point operation* $\Pi: S_o \times A_1^1 \times A_2^2 \times \dots \times A_n^n \rightarrow A_i^{n+1}$ induces a function π such that:

$$f_{n+1}(p) = \pi(f_1(\lambda, A_1^1), \dots, f_n(\lambda, A_n^n)), \quad \forall p \in S. \quad \square$$

For point operations, the value of the output attribute at each location is a function only of the input attribute values at the corresponding location. One example would be the boolean operations: “Calculate a soil aptitude map based on climate, soil, and slope maps, where the conditions are such that a soil is deemed “good for agricul-

ture” if it rains more than 1000 m/year and the soil has a *ph* between 6.5 and 7.5, and the slope is less than 15%”. This operation can be easily defined, based on the proposed atomic operations VALUE and ASSIGN (defined in section 4.1.1).

5 Conclusions and Future Work

In this paper, we have presented a generic definition for geographical objects. From this definition, we obtained different types of geographical entities and formulated a minimal set of general GIS operations. We have shown that this minimal set can be used to build more complex operations, equivalent to usual definitions of spatial selections and map algebra, but which are not constrained to any particular model of geographical data (such as the field or object models).

One of the most relevant aspects of this definition is its impact on the questions of interoperability. It has been recognized that interoperability in GIS requires a level of semantic modeling to account for the correspondence of concepts between different systems [10]. By establishing a formal notion and referring the concepts presented in GIS literature to this definition, we could provide a basis for a unified semantic framework for GIS.

This semantic framework can be materialized in the definition of a spatial language that can be completely representation-independent and define an interoperable framework for different GIS implementations. We intend to pursue this issue further in later works.

We point out that a unified formal definition of geographical objects does not, in any respect, diminish the relevance of the conceptual debate at the semantic level. The field and object views are, in fact, based on deeply rooted notions in human perception of his environment. Our primary aim was to indicate that the computerized modeling of the geographical data is necessarily a reductionism view, leading to formal notions which provides a basis for unified semantics of GIS operations.

We should point out that the concept of a geographical object introduced allows for new categories of spatial objects whose geometric support are not defined as a subset of the plane. This makes it very suitable to define, for instance, a concept of a volumetric geographical object, which seems to be quite adequate to bring volumetric visualization techniques into the realm of GIS systems. More generally, GIS systems will incorporate more and more multimedia techniques. The concept of geographical object is ready to absorb this tendency. We intend to discuss these issues in our future work.

Acknowledgements

This work has been partially motivated as part of the effort on the development of SPRING, a GIS built by INPE and its partners. In particular, the first author would like to thank his colleagues on the SPRING project, especially Ubirajara Freitas and Ricardo Cartaxo, for many a fruitful discussion on the topic of GIS modeling. This work has been partially financed by a joint NSF/CNPq project on "Interoperability on GIS" (grant CNPq 480322/99).

References

- [1] P. Burrough, A. Frank (eds). *Geographic Objects with Indeterminate Boundaries*. London, Taylor and Francis, 1996.
- [2] H. Couclelis. People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In *Theories and methods of spatio-temporal reasoning in geographic space*, eds. A. U. Frank, I. Campari, and U. Formentini, 65-77. Berlin: Springer Verlag.
- [3] H. Couclelis. From cellular automata to urban models: new principles for model development and implementation. *Environment and Planning B: Planning and Design*, 24(2): 165-74, 1997.
- [4] A. Requicha. Representations for Rigid Solids: Theory, Methods and Systems. *ACM Computing Surveys*, 12:437-464, 1980.
- [5] J. M. Gomes, L. Velho. Abstraction Paradigms for Computer Graphics. *The Visual Computer*, 11:227-239, 1995.
- [6] J. M. Gomes, B. Costa, L. Darsa, L. Velho. *Warping and Morphing of Graphical Objects*. Morgan Kaufman Publishers, San Francisco, 1998.
- [7] J. M. Gomes, B. Costa, L. Darsa, L. Velho. Graphical Objects. *The Visual Computer*, 12(6):269-282, 1996.
- [8] ESRI. *Arc/Info Data Management Concepts, data models, database design, and storage*. Redlands, CA, Environmental Systems Research Institute, 1994.
- [9] OpenGIS Consortium, *Technical Specifications*. <http://www.opengis.org>, Oct. 1998.
- [10] G. Câmara, R. Thomé, U. Freitas, A. M. Monteiro. Interoperability In Practice: Problems in

Semantic Conversion from Current Technology to OpenGIS. In: *2nd International Conference for Interoperability and Geographic Information Systems*. Zürich, Switzerland, March 1999.

- [11] D. Tomlin, *Geographic information systems and Cartographic Modeling*. Prentice Hall, New York, 1990.
- [12] M. Egenhofer. Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering*, 6:86-95, 1994.
- [13] M. Egenhofer, D. Mark, J. Herring. *The 9-Intersection: Formalism and its Use for Natural Language Spatial Predicates*. NCGIA Technical Report 94-1, May 1994.
- [14] A. U. Frank. Qualitative Spatial Reasoning about Distances and Directions in Geographic Space. *Journal of Visual Languages and Computing*, 3: 343-371, 1992.