SPRING and TerraLib: Integrating Spatial Analysis and GIS

GILBERTO CÂMARA[1], MARCOS NEVES[1], ANTÔNIO MIGUEL VIEIRA MONTEIRO[1],
RICARDO CARTAXO MODESTO DE SOUZA[1], JOÃO ARGEMIRO PAIVA[1], LÚBIA VINHAS[1]
[1] Instituto Nacional de Pesquisas Espaciais - INPE
Av. dos Astronautas, 1758, São José dos Campos (SP), Brazil 12227-001
{gilberto, marcos, miguel, cartaxo, miro, lubia}@dpi.inpe.br

Abstract. This work describes the development of a SPRING and TerraLib, two tools for integrating spatial analysis in a GIS environment. SPRING is an established freeware GIS, available on the Internet, which includes functions for geostatistics, point pattern analysis and exploration of area data. TerraLib a GIS library, a new development by INPE, that is aimed at providing a rich and powerful environment for the development of GIScience research. TerraLib is an open source software tool, allowing a collaborative environment and its use for the development of multiple GIS appliances.

Keywords. GIS, Spatial Analysis, Software Libraries.

1 Introduction

The development of tools for integrating spatial analysis and GIS has received much attention lately and is one of the main research areas of the NSF-funded CSISS (Center for Spatially Integrated Social Science)(Goodchild et al. 2000). Currently, most spatial analysis tools are linked to a GIS via loose coupling mechanisms, where the GIS is used for data conversion and graphic display, and the spatial models are run outside of the GIS structures. An example is the link between SpaceStat and ArcView (Anselin and Bao 1997). This structure allows use of existing programs, but requires substantial work in data conversion and causes problems of redundancy and consistency, due to the creation of multiple versions of the same data. Modelling tools also lack sufficiently flexible GIS-like spatial analytical capabilities; as a result, their ability to convey spatial relations is limited. By contrast, in a tight level of integration, there would be no strict distinction between model and GIS, and a spatial model becomes just one of the applications that could be constructed using the generic functionality of a GIS toolbox(Wesseling et al. 1996). A strongly-integrated GIS and spatial analysis architecture would allow non-specialists, already familiar with GIS interfaces, to experiment with models, reducing the overhead for data conversion and abstracting part of the complexities in model formulation.

The integration challenge should not be minimised, since the end product should not be too specialised as to have a steep learning curve, neither be user friendly but not powerful enough to satisfy the demands of the academic market (Goodchild et al. 2000). Additionally, GIS commercial companies may lack enough market incentive to enhance their product line with spatial analytical tools. Companies naturally prefer to invest on techniques with an established theory and a proven record of applications, such as the recent inclusion of kriging techniques in major commercial systems (Krivoruchko 2001). A further complication is the emphasis of most commercial systems on application development tools based on Microsoft's Visual Basic? , which is inadequate to be used as a basis for supporting spatial analytical functions. This situation is in sharp contrast with the need of the spatial analysis community to share experiences and results, and to experiment with new techniques. To cater for this need, CSISS has proposed the development of a suite of modular software components to carry out spatial statistical and spatial econometric analyses(Goodchild et al. 2000). Actually, this situation is part of a more general problem, in that the GIScience community currently lacks a comprehensive set of open-source tools for development of new ideas and rapid prototyping. In an ideal situation, the spatial analytical community would have access to a set of tools for rapid GIS application development, where new algorithms could be tested, within a user-friendly GIS environment, which would include data conversion, data management and visualization functions.

To face this challenge, this work discusses the development of two tools for integration of spatial analysis functions in a GIS environment. The first of these tools, SPRING, is a GIS which has been developed over the last 8 years, and is available on the Internet as a freeware tool. SPRING has been enhanced in recent years to include functions for geostatistics, point pattern analysis and exploration of area data. In section 2 of this paper, we present a brief discussion on SPRING and indicate some of the spatial analysis applications that have been developed using this system. Based on the lessons learned in the SPRING development, and aiming to support the development of open source tools for spatial analysis, the authors are developing **TerraLib,** an open-source general-purpose GIS application development library. **TerraLib** provides functionality for handling the different types of geographical data and facilities for data conversion, graphical output, and spatial database management using a spatially-enabled object-relational DBMS (such as ORACLE and PostGIS). In section 3 below, we describe TerraLib in more detail and in section 4, we describe TerraView, a tool for exploratory spatial data analysis (ESDA) based on TerraLib. We conclude the paper by examining the possible impact of the TerraLib project in the GIScience community.

2 SPRING

In 1992, INPE started the development of SPRING, whose first Internet version was made available in late 1996. Now in version 3.5, SPRING provides a comprehensive set of functions for processing of spatial information, including tools for Satellite Image Processing, Digital Terrain Modeling, Spatial Analysis, Geostatistics, Spatial Statistics, Spatial Databases and Map Management. SPRING has required over 170 man/years of development and includes extensive documentation, tutorials and examples. It is arguably the most powerful free GIS available on the Web (http://www.dpi.inpe.br/spring). More than 21,000 users from 60 countries have downloaded the software, as of May 2002. Versions exist for Windows 95/NT and Linux. In terms of spatial analysis, SPRING provides functions for geostatistics, point pattern analysis and exploration of area data, as described below.

2.1 Geostatistics in SPRING

The geostatistical module in SPRING is an interactive front-end to the GSLIB software library (Deutsch and Journel 1998) and includes the following capabilities:

?? *Exploratory data analysis*: descriptive statistics, and graphical tools, such as histograms, cumulative distribution functions, normal probability curves and scatterplots.

?? *Data transformation*: functions to filter outliers and to perform normal and logarithm transformations.

?? *Data modeling:* interactive construction of linear and surface variograms; model fitting and estimation of the correlation structure. Construction of indicator variables and associated variograms.

?? *Data inference*: interpolation by ordinary and indicator kriging (both for categorical and numerical variables). Uncertainty estimation by indicator kriging (Goovaerts 1997).

?? *Data validation*: Cross validation procedures and functions for error and residues analysis.

We used geostatistical techniques to produce trend surfaces for the homicide rates in São Paulo in 1996 and 1999, as shown in Figure 1. The original data consisted of estimates of homicide rate per 100,000 inhabitants, aggregated by the 96 districts of the city. The data is approximately gaussian, and the Shapiro-Wilk normality test indicates a value of 0,9653 (p-value = 0,012) To produce these maps, we obtained a sample set by assigning a sample at the center of each district. These samples were then used as a basis for computing a variogram that models the spatial correlation structure, and a surface was interpolated by ordinary kriging. The trend surfaces depict a significant decrease in the areas with lowest homicide rate (below 30 deaths per 100,000 inhabitants) in 1999 in relation to 1996. Since the lower homicide rate correlates very strongly with the wealthiest regions of the city, these results indicate a spatial spreading of crime. Violence is thus not confined to the poorest areas of the city and the inhabitants of richer areas are increasingly prone to be victims of violent assaults.

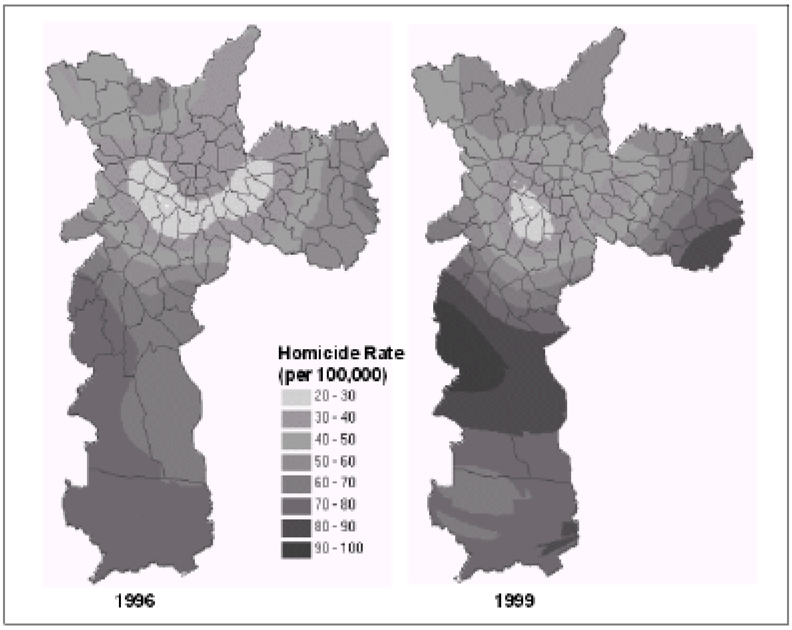

Figure 1 – Estimated surfaces for homicides in São Paulo, with values shown as homicides per 100,000 population. Left: 1996 data Right: 1999 data.

2.2 Point Pattern Analysis in SPRING

The point pattern analysis functions in SPRING include first-order density non-parametric kernel density estimation (Silverman 1986) and second-order dependence estimation by means of nearest neighbor and Ripley's K-function (Ripley 1981). As an example, consider the case of the data shown in Figure 2, that depicts the location of the events related to infant mortality in the Brazilian city of Porto Alegre. This data were used to produce a surface showing relative intensity of infant deaths, using non-parametric *Kernel estimation* (Figure 3) and to estimate the extent to which there is spatial dependence in the arrangement of events, using the L-function (figure 4), which is the difference between the measured K-function on the data and estimated K-function in the case of a random point pattern (Bailey and Gattrel 1995). In the latter case, the software also computed the envelope obtained a Monte Carlo randomization of events., which shows a strong spatial dependence within the distance range of 0-6 km.

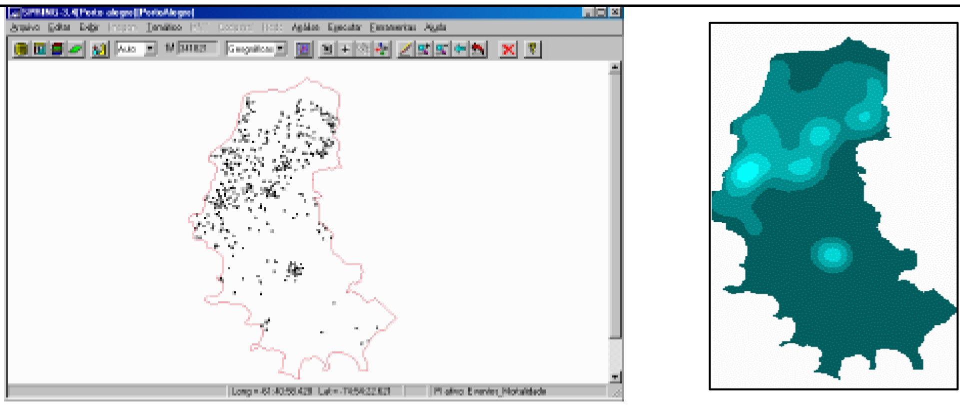

Figure 2 Distribution of infant deaths cases in Porto Alegre in 1996 (left) and non-parametric kernel estimated intensity surface (right).

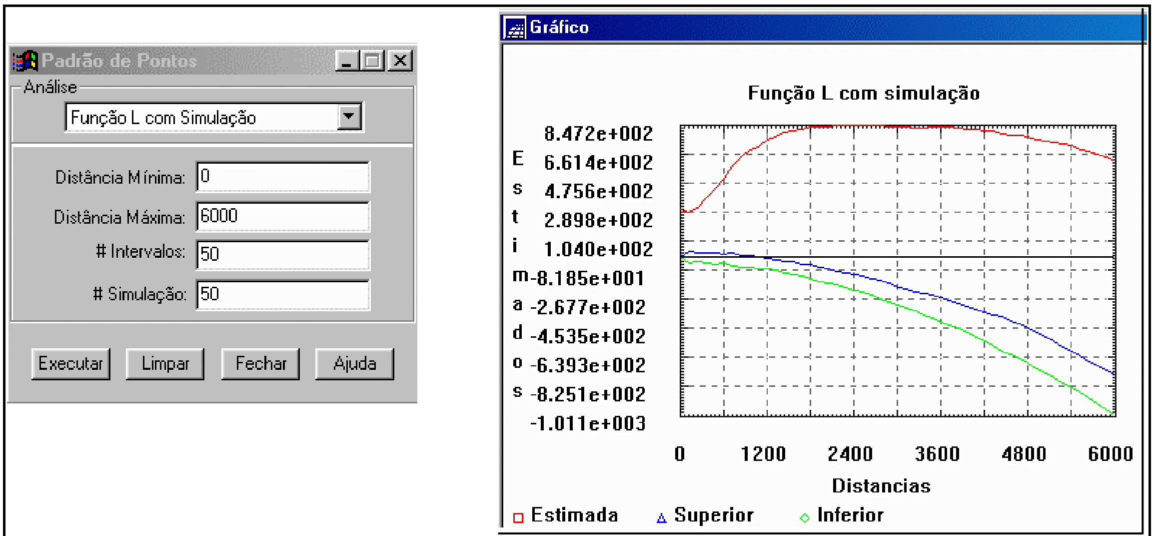

Figure 3 – L-function for infant death cases in Porto Alegre in 1996 (in red) with simulated Monte Carlo envelopes (maximum = blue; minimum=green).

2.3 Analysis of Area Data in SPRING

The spatial analysis tools for area data in SPRING include: (a) *visualization*: attribute and spatial queries and thematic map generation; (b) *exploration*: global and local Moran autocorrelation indices; (c) *manipulation*: LEGAL map algebra language. A link to SpaceStat? is also included . An exemple is shown in Figure 4, which depicts the LISA map (Anselin 1995) and the Moran map (Anselin 1996) for the social exclusion index of São Paulo (1991 census). This data is part of a study on the use of spatial analytical tools for finding spatial patterns in social exclusion indicators (Câmara et al. 2002).

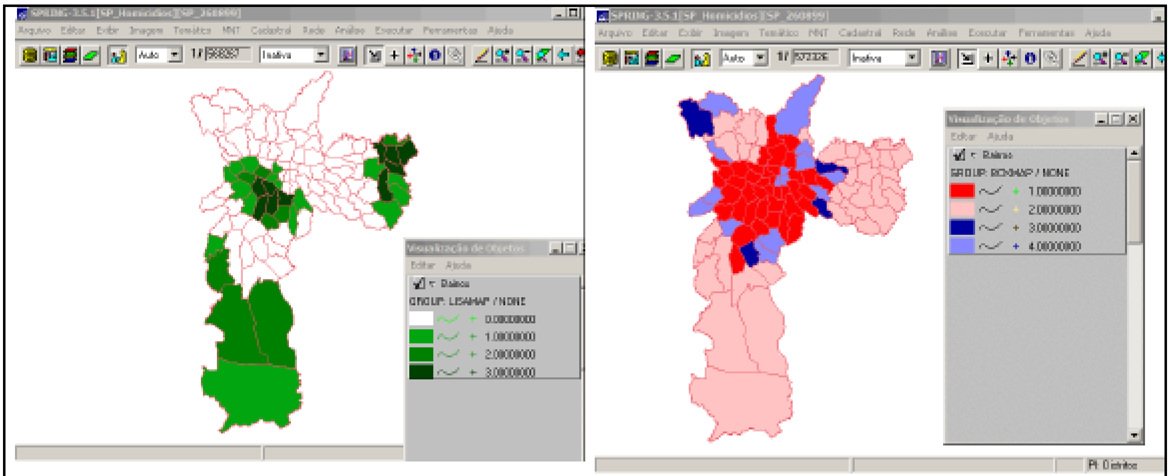

Figure 5 - Right: LISA local spatial autocorrelation index for the social exclusion index in São Paulo (1991 census) (1= 95% significance; 2= 99% significance; 3= 99.9% significance.

Left: Moran map for social exclusion index in São Paulo (1991 census). Quadrant 1 (high values, high neighbors) is shown in red; Quadrant 2 (low values, low neighbors) is shown in pink; Quadrant 3 (high values, low neighbors) is shown in dark blue; Quadrant 4 (low values, high neighbors) is shown in light blue.

3 The TerraLib GIS Library

3.1 General Description

GIS software development is bound to witness substantial change in the upcoming years, induced by technological advances in spatial databases. Current and expected advances in database technology will enable, in the next few years, the complete integration of spatial data types in data base management systems. This integration is bound to change completely the development of GIS technology, enabling a transition from the monolithic systems of today (that contain hundreds of functions) to a generation of *spatial information appliances*, small systems tailored to specific user needs (Egenhofer 1999). Motivated by the database paradigm shift and backed by its previous experience with SPRING, the INPE group is currently developing **TerraLib**, an open-source GIS component library for developing GIS applications, where all spatial data (raster + vector) is stored in a DBMS, such as Oracle9i, Postgres (Postgis), MySQL. TerraLib is developed in C++, and the library has been divided in three components:

?? **kernel**: composed of classes for storing geometries and attributes in an object-relational DBMS, such as ORACLE and PostgreSQL, cartographic projections, and topological and directional operators. Kernel maintenance and upgrade is the responsibility of the project core team, as typical for other free software projects.

?? **functions**: algorithms that use the kernel structures, including spatial analysis, query and simulation languages, and data conversion procedures. Again, maintenance and upgrade is the responsibility of the project core team, but it is expected that new functions developed by external collaborators will be incorporated.

?? **contrib**: applications built by users of TerraLib, including external authors, who are responsible for their maintenance.

3.2 Generic Programming in TerraLib

The object-oriented paradigm has been adopted as the paradigm of choice in most software development architectures, including most of the tools available for GIS development. This paradigm is based on the principle of grouping similar elements of the world into classes and making commonality explicit by means of inheritance mechanisms. By using inheritance, derived classes share common properties of a base class, while maintaining their specialized behaviour (Stroustrup 1997). In GIS software development, this paradigm often leads to the establishment of classes, which are directly related to the different types of geometrical representation. For example, a typical object-oriented GIS might have basic classes such as Image, Polygon, Polygon Map, Grid, Tin, Point, and Point Set, containing both the underlying data structure and the corresponding set of algorithms. This class subdivision helps software developers by breaking the system into modules and allowing a convenient division of labour amongst the development team.

There is, however, a drawback to the straightforward object-oriented approach to GIS software development: the attachment of algorithms to data structures. Algorithms form a basic core of most successful GIS, and a large number of them do not depend on some particular implementation of a data structure but only on a few fundamental semantic properties of the structure (Austern 1998). Such properties can be - for example - the ability to get from one element of the data structure to the next, and to compare two elements of the data structure. For example, for an algorithm for computing spatial autocorrelation indexes it is not essential if the elements are organised as a set of points a set of polygons, a TIN, a grid or an image. All that is needed is the ability to look into a list of values, and to obtain, for each element of the list, its values and the indexes of the elements of the list that satisfy a certain property (for example, those that are closer in space than a specified distance). In a similar way, a large number of spatial analysis algorithms can be abstracted away from a particular data structure and described only in terms of their properties. By contrast, in many GIS libraries, the misuse of object-oriented principles has resulted in classes that contain both the underlying data structure and the corresponding set of algorithms; the algorithms are unnecessarily linked to a particular type of data structure, and the same algorithm (for example, the calculation of a histogram) is implemented separately for each spatial data structure.

The separation of algorithms from the data structures is the basis of the *generic programming* (Austern 1998). Generic programming is based on the idea that there are fundamental laws that govern the behaviour of software components and designs interoperable modules based on these laws. This idea has led to programming tools such as the STL (Standard Template Library), which is part of the ISO C++ standard (Stroustrup 1997), and provides a large number of generic containers (such as *list*, *set* and *map*) as well as algorithms (such as *sort* and *search*) whose behaviour is independent of the containers. The key for the separation between containers and algorithms is the use of *iterators*, which are generalized pointers that provide a glue for connecting algorithms and data structures. Generic programming has emerged as such an relevant programming paradigm that many software engineering researchers are recommending a development approach called *multiparadigm programming,* that proposes the application of different styles of programming, where they are most appropriate (Coplien 1999). In the case of algorithms, whose behaviour is independent of the data structures, Coplien (1999) proposes the use of *generic programming,* implemented as *templates* in C++, as better suited to express their commonality properties than object-oriented programming.

The application of the generic programming paradigm to GIS is a four-step process: (a)finding regularities in spatial data handling algorithms and data structrures; (b) generalising the regularities in data structures into abstract types (containers); (c) providing iterators that allow access to the containers; (d) designing algorithms that use these iterators. These ideas have been used for the design of TerraLib's spatial analysis algorithms, which are designed to work on a variety of spatial data types.

3.3 TerraView: Spatial Analysis based on TerraLib

One of the most important aims of the TerraLib project is to support the development of a new GIS tool for spatial data analysis. This product is called **TerraView**, an open source product, which aims at covering a wide range of needs for spatial data analysis, including the following characteristics:

?? *Spatial Data Support:* the software will support a variety of different types of spatial data, including surfaces, images, polygons, networks and points and their descriptive attributes. Provision will also be included to support a temporal dimension associated with any one of these data types.

?? *Data Conversion*: efficient data translation tools allowing users to import and export data from or into other existing software systems. The product should also allow conversion of tabular data, which has been, assigned a geographical reference (as a census tract number or a zipcode). Supported formats include: ARC/VIEW, MapInfo, InfoMap, DBASE, SPRING, TABWIN, and R.

?? *Data Integration*: a range of GIS and related tools to allow data layers of different types (surfaces, points etc.) to be readily combined to the same spatial reference frames.

?? *Geostatistics and Uncertainity modelling: all* geostatistical software available in SPRING (see section 2) will be included in SPRINGExplorer, plus additional functions for sequential simulation support.

?? *Interactive Visualisation tools*: These tools allow data to be explored simultaneously from several different perspectives (table, scatterplot, and map), using several windows.

?? *Exploratory spatial data analysis (ESDA)*: global and local area spatial statistics, heuristic search, and various forms of pattern detection algorithms and non parametric models.

?? *Spatial Statistical modelling techniques:* support for regression analysis in the presence of spatial effects, specifically spatial heterogeneity and spatial dependence. Includes techniques to detect spatial and non-spatial clustering.

A preview of the initial version of TerraView is shown in Figure 6.

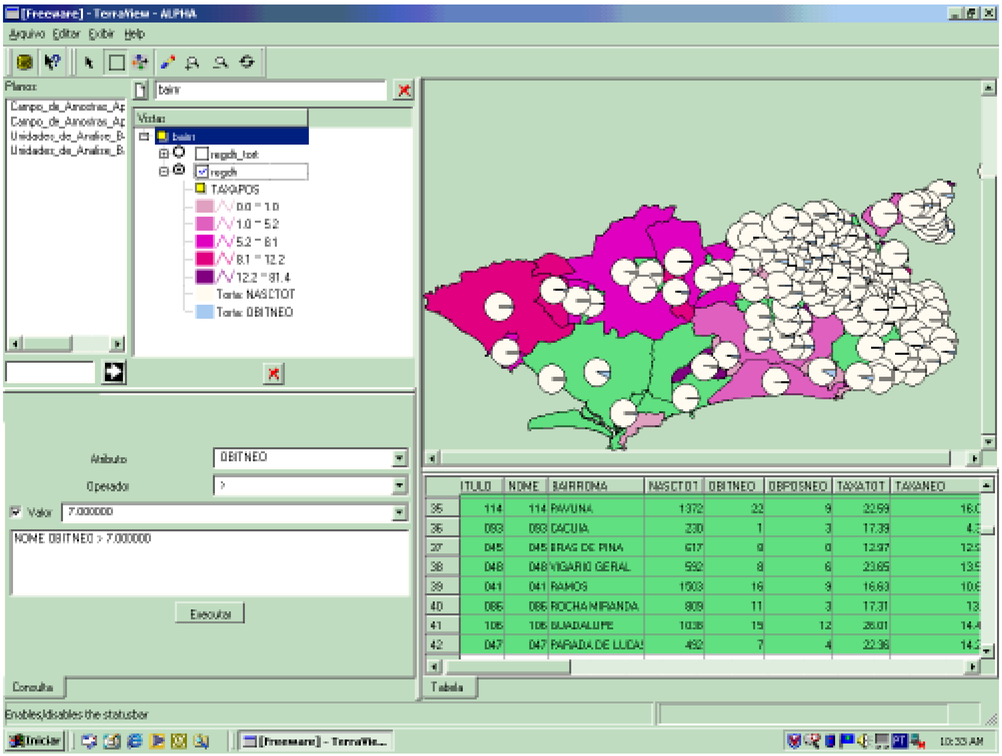

Figure 6 – Initial Version of User Interface for the TerraView product

4 Conclusion

The integration of GIS and spatial analysis tools in SPRING has proven extremely important and has added substantial value to the system. A large number of users, especially on the environmental and health areas in Brazil, are developing applications based on such tools. The authors consider that the geographical information community would benefit from the availability of a general, open source GIS library, such as TerraLib, and an open source spatial analytical tool, such as TerraView. In a similar approach to the Linux and subsequent open source software efforts, we recognize that collaborative development does not happen by spontaneous growth. There must be created a core set of technologies from which further developments may freely extend. We hope that TerraLib and TerraExplorer will have a positive impact, by allowing researchers and solution developers to share their expertise and ideas. As of late April 2002, the status of these projects is that TerraLib will have its first public release in May 2002 and TerraView in July 2002.

Acknowledgments

TerraLib is large team development effort, being developed by INPE in cooperation with TECGRAF/PUC-RIO (The Computer Graphics Group at the Catholic University in Rio). The library core team, apart from the authors, include Karine Ferreira, Gilberto Ribeiro, Juan Garrido, Lauro Hara at INPE and Marcelo Tilio de Carvalho and Paula Frederick at PUC-RIO.

The author's research is supported by a joint CNPq (Brazil)/NSF (USA) grant on "Computational Issues in Interoperability in GIS" (award number CNPq 480322/99, by Brazil's MCT/PCTGE (Ministry of Science and Technology—Program for Science and Technology on Ecosystems Management) and by the Foundation for Research in São Paulo - FAPESP - through its program for Public Policy Research (grant 00/01965-0).

References

Anselin, Luc. 1995. Local indicators of spatial association - LISA. *Geographical Analysis* 27:91-115.

———. 1996. The Moran scatterplot as ESDA tool to assess local instability in spatial association. In *Spatial Analytical Perspectives on GIS*, edited by M. Fisher, H. J. Scholten and D. Unwin. London: Taylor & Francis.

Anselin, Luc, and Shuming Bao. 1997. Exploratory spatial data analysis linking SpaceStat and ArcView. In *Recent developments in spatial analysis*, edited by M. Fischer and A. Getis. Berlin: Springer-Verlag.

Austern, Matt. 1998. *Generic Programming and the STL : Using and Extending the C++ Standard Template Library*. Reading, MA: Addison-Wesley.

Bailey, Trevor, and Anthony Gattrel. 1995. *Spatial Data Analysis by Example.* London: Longman.

Câmara, Gilberto, Antonio Miguel Monteiro, Frederico Ramos, Aldaiza Sposati, and Dirce Koga. 2002. Mapping Social Exclusion/Inclusion in Developing Countries: Social Dynamics of São Paulo in the 90's. In *Spatially Integrated Social Science: Examples in Best Practice*, edited by M. Goodchild and D. Janelle. Santa Barbara, CA: Center for Spatially Integrated Social Science.

Coplien, James. 1999. *Multi-Paradigm Design for C++*. Reading: Addison-Wesley.

Deutsch, Clayton, and André Journel. 1998. *GSLIB: Geostatistical software library and user´s guide.* New York: Oxford University Press.

Egenhofer, Max. 1999. Spatial Information Appliances: A Next Generation of Geographic Information Systems. Paper read at First Brazilian Workshop on GeoInformatics, at Campinas, Brazil.

Goodchild, Michael, Luc Anselin, Richard Applebaum, and Barbara Harthorn. 2000. Toward Spatially Integrated Social Science. *International Regional Science Review* 23 (2):139-159.

Goovaerts, Pierre. 1997. *Geostatistics for Natural Resources Evaluation*. New York: Oxford Univ. Press.

Krivoruchko, Konstantin. 2001. Using linear and non-linear kriging interpolators to produce probability maps. Paper read at IAMG2001 Conference of the International Association for Mathematical Geology, at Cancun, Mexico.

Ripley, Brian. 1981. *Spatial statistics.* Chichester: John Wiley.

Silverman, B.W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Stroustrup, Bjarne. 1997. *The C++ Programming Language, 3rd ed.*: Addison-Wesley Publishing Company.

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.